Artificial Intelligence Review (2024) 57:260
https://doi.org/10.1007/s10462-024-10888-y

Large language models (LLMs): survey, technical frameworks, and future challenges

Pranjal Kumar[1]

Accepted: 25 July 2024 / Published online: 18 August 2024

Abstract
Artificial intelligence (AI) has significantly impacted various fields. Large language models (LLMs) like GPT-4, BARD, PaLM, Megatron-Turing NLG, Jurassic-1 Jumbo etc., have contributed to our understanding and application of AI in these domains, along with natural language processing (NLP) techniques. This work provides a comprehensive overview of LLMs in the context of language modeling, word embeddings, and deep learning. It examines the application of LLMs in diverse fields including text generation, vision-language models, personalized learning, biomedicine, and code generation. The paper offers a detailed introduction and background on LLMs, facilitating a clear understanding of their fundamental ideas and concepts. Key language modeling architectures are also discussed, alongside a survey of recent works employing LLM methods for various downstream tasks across different domains. Additionally, it assesses the limitations of current approaches and highlights the need for new methodologies and potential directions for significant advancements in this field.

Keywords Generative language models · Artificial intelligence · Natural language processing · Machine learning · Neural networks · Large language models

1 Introduction

The progression of NLP and AI models has traced a significant path, commencing with rule-based systems circa the mid-1990s, shifting to statistical models by the late 1990s, and ultimately progressing to neural networks in the early 2000s (Ling et al. 2023). The implementation and success of RNN-based "self-attention" and "Transformer-based" neural network architectures (Vaswani et al. 2017) have significantly contributed to the increased prevalence of pre-trained language models (PLMs) during the late 2010s. These PLMs are capable of learning universal language representations from extensive datasets without human intervention. This form of unsupervised learning is particularly advantageous for various downstream NLP tasks, including answering multiple-choice

✉ Pranjal Kumar
pran04146@gmail.com

[1] Department of Intelligent Systems, School of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab 144411, India

questions (Robinson et al. 2022), story generation (Cao et al. 2023), and common sense reasoning (Yang et al. 2023). Additionally, they mitigate issues related to overfitting. In recent years, there has been significant progress in the development of LLMs (Wei et al. 2022), as demonstrated by GPT-3 (OpenAI) (Brown et al. 2020), PaLM (Google) (Chowdhery et al. 2022), and LLaMA (Meta) (Touvron et al. 2023), Megatron-Turing NLG (NVIDIA) (Smith et al. 2022), and Jurassic-1 Jumbo (AI21 Labs) (Reed et al. 2022), among others. The observed expansion can predominantly be credited to the exponential augmentation in extensive datasets and advancements in computational hardware capabilities. Scholars have noted that augmenting the model's size and the quantity of training data consistently augments the model's capability, conforming to a scaling law (Kaplan et al. 2020). LLMs have emerged as a significant area of AI research due to their superior performance in understanding and generating human-like text compared to smaller models. LLMs possess the capacity to revolutionize both scientific and social sciences by accelerating research, enhancing the process of discovery, and fostering interdisciplinary collaboration. This is achieved through streamlined literature analysis, creative idea generation, and intricate data interpretation.

The capabilities of LLMs as versatile problem-solving tools have led to their expanded applications beyond simple chatbots (OpenAI 2023). They are now being utilized as assistants or even replacements for human workers or traditional tools in industries such as healthcare, banking, and education. These capabilities are starting to assume significant roles, particularly within the healthcare sector. For instance, GPT-4 is used to transcribe medical dictations directly into electronic health records (EHRs). This task, typically done by medical scribes, is being increasingly automated. Tools like Nuance's Dragon Medical One (Onitilo et al. 2023) use LLM technology to accurately convert speech to text, allowing doctors to focus more on patient care and less on paperwork. LLMs can process and summarize vast amounts of medical literature quickly (Watanabe and Wiseman 2023), a task often done by research assistants. Tools like Iris.ai use AI to help researchers find and summarize relevant scientific papers, thus speeding up the research process and reducing the need for human labor in literature review and synthesis. However, the direct application of LLMs to domain-specific problems presents significant challenges. Firstly, there is a substantial difference in the types of speech and language used in various contexts, such as a doctor's office, a courtroom, and online discussions. Even for humans, acquiring such skills and expertise requires extensive training, much of which is hands-on and confidential. Additionally, a single generic LLM solution cannot readily replace the "business models" that individual fields, institutions, and teams use to maximize their utility functions for specific activities. Furthermore, the professional use of LLMs requires in-depth, real-time, and accurate domain knowledge that pre-trained LLMs cannot easily provide.

The development of artificial intelligence has significantly transformed various sectors. Recently, the field of NLP has seen substantial progress, with LLMs becoming a prominent area within it (Sarker 2022). LLMs are capable of generating human-like language and performing a range of language processing tasks due to their training on extensive textual datasets, which include publicly available data and data licensed from third parties. Notable examples include the generative pretrained transformer (GPT) from OpenAI and BARD from Google (Michael 2020; Deng and Lin 2022; Jordan and Mitchell 2015). NLP is a critical aspect of artificial intelligence that focuses on the interactions between machines and humans for communication. Various sectors such as management, finance, retail, law, architecture, and transportation can leverage enhanced computer capabilities in understanding and manipulating human language (Brown et al. 2020). The advancements in LLMs have greatly enhanced our comprehension and application of AI in these fields, and their influence continues to grow, significantly impacting everyday life.

Advancements in AI are expected to further impact the future of learning and discovery significantly. For example, the latest GPT model, GPT-4, incorporates enhanced features such as increased safety, multilingual support, text generation from images, and tools for drug discovery, which are not present in earlier versions like GPT-3 and GPT-3.5. Despite its capabilities, GPT-4 has certain limitations. These include hallucinations-production of text by a language model that appears plausible but contains information that is either incorrect, not based on the model's training data, or is contextually inappropriate. This phenomenon occurs due to the model's probabilistic nature, which predicts the next word or phrase based on learned patterns rather than a strict verification of factual accuracy (Manakul et al. 2023). Additionally, challenges such as data privacy, algorithmic bias, and the ethical implications of AI-driven decision-making need to be addressed when implementing AI technologies (Korteling et al. 2021; Sarker 2022). To fully leverage AI's potential for enhancing learning and scientific discovery, these challenges must be resolved and ethical guidelines established. Although large language models have made significant progress in recent years, they still require further development to become truly useful. A major limitation is their lack of interpretability, which hampers understanding the rationale behind the model's predictions. Ethical considerations include potential risks such as improper use, unethical implementation, compromised integrity, and various other concerns. Overall, while large language models continue to push the boundaries of natural language processing, significant efforts are needed to address their limitations and the related ethical issues.

1.1 Organisation of paper

The structure of this study is as follows: Sect. 2 provides a comprehensive analysis and comparison of results from various recent extensive surveys on LLMs. Section 3 explains the technical aspects related to the framework and structure of LLM systems, while Sect. 4 investigates the word embeddings used in LLMs. The overall taxonomy of the paper is illustrated in Fig. 1. Subsequently, Sects. 5 and 6 explore relevant research conducted across multiple fields. These sections cover a broad range of classifications and methods based on LLMs applicable in diverse settings and domains. The concluding section of the paper addresses the most contentious topics and their potential future development, summarizing the discussion.

2 Prior works

Previous surveys relevant to LLM specialization are concisely reviewed in this section.

Recent review articles have deliberated on the advantages and indispensability of tailoring LLMs for particular domains. The critical risks of using generic LLMs in specialized fields such as medical education are emphasized in Sallam (2023), primarily due to their lack of specificity and precision. Additionally, Shaghaghian et al. (2020) provides recommendations for implementing language models tailored to the legal domain. Initial research on a finance-focused LLM has displayed encouraging outcomes, showcasing enhanced efficacy in financial assignments while maintaining parity with overall benchmarks (Wu et al. 2023). Given these developments, it is essential

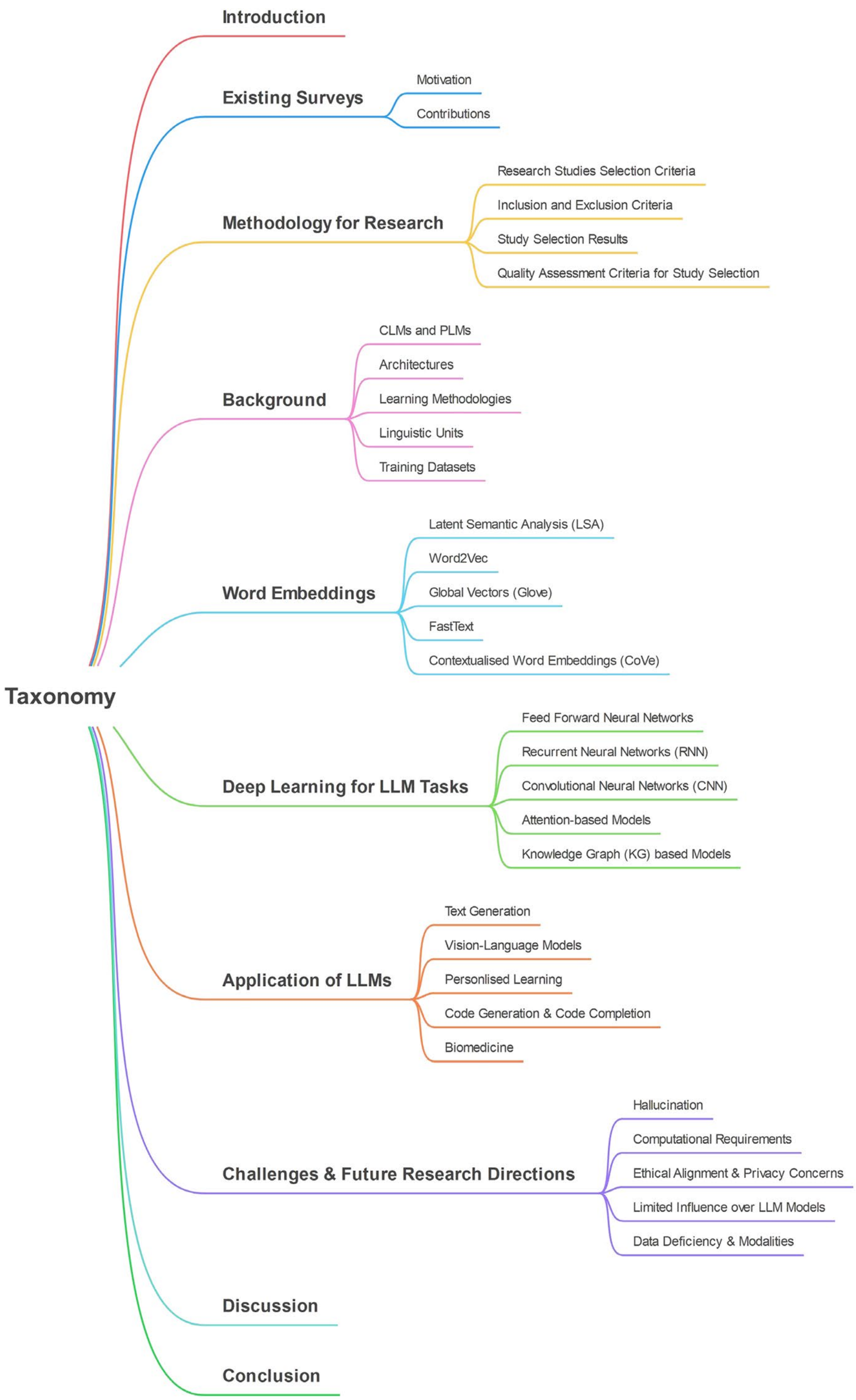

Taxonomy
Introduction
Existing Surveys
Motivation
Contributions
Methodology for Research
Research Studies Selection Criteria
Inclusion and Exclusion Criteria
Study Selection Results
Quality Assessment Criteria for Study Selection
Background
CLMs and PLMs
Architectures
Learning Methodologies
Linguistic Units
Training Datasets
Word Embeddings
Latent Semantic Analysis (LSA)
Word2Vec
Global Vectors (Glove)
FastText
Contextualised Word Embeddings (CoVe)
Deep Learning for LLM Tasks
Feed Forward Neural Networks
Recurrent Neural Networks (RNN)
Convolutional Neural Networks (CNN)
Attention-based Models
Knowledge Graph (KG) based Models
Application of LLMs
Text Generation
Vision-Language Models
Personlised Learning
Code Generation & Code Completion
Biomedicine
Challenges & Future Research Directions
Hallucination
Computational Requirements
Ethical Alignment & Privacy Concerns
Limited Influence over LLM Models
Data Deficiency & Modalities
Discussion
Conclusion

Fig. 1 Taxonomy of this review

to undertake a thorough examination and precise categorization of domain specialization methods to facilitate the successful application of LLMs across various industries. Research into effective and efficient adaptation of PLMs to different domains encompasses methods such as introducing new model layers or adjusting model parameters, as detailed in surveys (Ding et al. 2022; Guo and Yu 2022). The review (Reis et al. 2021) is one of the most current and relevant surveys of deep learning models that utilize transformers as their core approach for language understanding. It reviews addresses knowledge-encoding strategies for these models and highlights issues such as reliance on context and language. For optimal performance and efficiency in standard NLP tasks, the survey (Zhang and Yang 2018) summarizes and analyzes existing NLP models. The primary value of this survey lies in its detailed information on various architectures and their functionalities. However, LLMs do not benefit from these methods due to the complexity and inaccessibility of their architecture and parameter space. The computational demands and need for effective optimization strategies pose challenges to maintaining the expertise of LLMs.

Several systematic literature reviews have focused on specific applications of NLP and LLMs, such as automated feedback, chatbots, question generation, and essay scoring, to conduct their analyses. For example, Kurdi et al. (2020) performed a comprehensive evaluation of empirical studies addressing the issue of automatic question generation in educational settings. They provided an extensive overview of various generation strategies, tasks, and evaluation techniques described in the existing literature. Semantic-based question generation closely related to source content has significant potential for LLMs. The use of chatbots in educational contexts has been thoroughly investigated by Wollny et al. (2021). Their conclusion was that considerable effort is needed to fully harness the potential of chatbots, including improving their adaptability across diverse educational environments. Automatic essay scoring systems have been examined in a comprehensive literature review (Ramesh and Sanampudi 2022), highlighting the limitations of current systems based on classical machine learning (ML) and deep learning (DL) techniques. Some authors, such as in Mialon et al. (2023), discuss methods to enhance the reasoning capabilities of LLMs, while others, like in Yang et al. (2023) and Zhao et al. (2023), explore the foundations and potential applications of generative artificial intelligence. However, these studies did not fully detail the framework of LLMs. Overall, these extensive literature reviews have identified issues that could be addressed by implementing advanced LLMs (such as GPT-3 or Codex).

The previous research exhibits several limitations, which are detailed as follows:

- Although there are comprehensive analyses (Min et al. 2021; Qiu et al. 2020) of Pretrained Foundation Models (PFMs) and their application to various NLP tasks, these analyses may not be directly applicable to LLMs due to inherent differences.
- Numerous review articles have surfaced, focusing on distinct facets of LLMs, in response to their growing significance and established efficacy.
- Some studies Yang et al. (2023) and Zhao et al. (2023) address elements such as enabling reasoning capabilities in LLMs or examining the foundations and potential applications of generative artificial intelligence. However, these studies may not fully capture the detailed framework of LLMs.
- The systematic analysis and classification of LLM subfields have not been adequately addressed. Identifying the key subfields within the LLM domain and understanding their contribution to the development of general-purpose LLM frameworks is a recognized gap.

This study thoroughly investigates the methodologies employed in constructing general-purpose LLM frameworks, along with current trends and challenges in this domain.

2.1 Motivation

At present, there is no exhaustive and systematic analysis for categorizing LLM subfields within the context of multi-task NLP that includes explicit benefits and comparative analysis. This study rigorously investigates the techniques used in developing versatile LLM frameworks, as well as the current trends and challenges in this domain. The following enumerates the research questions.

1. What are the fundamental theoretical principles of LLMs and how do they facilitate advancements in natural language comprehension?
2. What impact do different design strategies in language modeling and word embeddings have on the performance and capabilities of LLMs?
3. What are the primary factors affecting the performance of LLMs in downstream tasks across various domains, and how do different LLM architectures perform in these contexts?
4. What are the distinctive features of major LLM architectures, and how do these features influence their performance in different applications?
5. What are the current unresolved issues and limitations in the domain of LLMs, and how can these issues be addressed to advance language comprehension?
6. What ethical factors necessitate examination during the advancement and implementation of LLMs, and what strategies exist to alleviate potential biases and ethical dilemmas?

2.2 Contributions

Recent research has extensively investigated methods for domain specialization in LLMs. Numerous conventional methodologies prioritize the development of universally applicable technical resolutions, capable of accommodating diverse domains with marginal adjustments and access to pertinent domain-specific data. However, there is currently no comprehensive standardization or summary of methodologies for evaluating different domain specialization strategies, making it difficult to cross-reference them across different application domains. This lack of transparency obscures the current bottlenecks, difficulties, unresolved problems, and potential future research areas, posing a challenge for non-AI specialists. This survey provides an in-depth and systematic analysis of the most recent advancements in LLM domain specialization. The following key points highlights the contributions of this work:

- A comprehensive introduction and context of LLMs is provided to facilitate understanding of advanced concepts.
- The design of LLMs, with an emphasis on language modeling and word embeddings, is thoroughly examined to improve understanding of various methodologies.
- An extensive evaluation of recent studies utilizing LLMs for various downstream tasks across different domains is conducted. Additionally, a summary of notable LLM architectures is included.

- The study has identified several unresolved issues in this field and discussed the potential future directions for LLMs.

3 Methodology for research

This work adhered to the guiding principles outlined by Kitchenham and Charters (Keele et al. 2007). Table 1 illustrates the use of PIOC techniques in developing the research questions.

3.1 Criteria for selecting research studies

Key phrases were chosen to obtain the necessary search results for exploring the research questions within the field. The search string used is: ('LLM' OR 'LLM Architectural features' OR 'Generative Language Models' AND 'Artificial Intelligence' OR 'Neural Networks' OR 'deep learning' AND 'GPT').

Table 2 exhibits the outcomes of the search. Although examination in this domain has persisted since 2000, the focus lies on scrutinizing papers published from 2014 to 2023 to depict the most recent progressions in the domain.

3.2 Inclusion and exclusion criteria

Only research papers that are pertinent are considered for this study. The studies encompassed various aspects, including refining the methodologies, examining the frameworks for LLM, and addressing diverse fields of application. The selected language is English, and all the items are subject to peer review.

3.3 Study selection results

A search query was selected to identify and classify 248 papers from the databases specified in Table 2. Following the removal of duplicate entries and the application of inclusion and exclusion criteria, 49 research papers were identified as appropriate for this study. This number was increased to 61 by integrating significant contributions using snowballing techniques. Subsequently, employing quality assessment standards, a total of 53 studies were selected for comprehensive analysis on Language Model Models (LLMs). The inclusion and exclusion criteria are detailed in Table 3.

3.4 Criteria for evaluating the quality of studies for selection

Quality assessment criteria are employed to ascertain the relevance of research papers in addressing the research questions. The research studies were assessed and given a score of either 1 or 0 according to the criteria specified in Table 4. This study considers a quality score of 4.

Table 1 Information regarding the PIOC

Research question	Population (P)	Intervention (I)	Outcome (O)	Context (C)
RQ1	Researchers, developers, users of LLMs	Theoretical analysis, Exploration of architectural elements	Advancements in natural language understanding, Core principles of LLMs	Contextual use-cases, Real-world applications of LLMs
RQ2	LLM designers, researchers, implementers	Variation in design methodologies, Word embedding techniques	Performance metrics, Functional capabilities of LLMs	Task-specific contexts, Data characteristics
RQ3	Downstream task practitioners, LLM developers	Factors affecting LLM performance, Architectural differences	Task-specific outcomes, Comparative performance metrics	Domain-specific requirements, Dataset diversity
RQ4	Researchers, LLM architects	Architectural features, Unique characteristics of LLMs	Application-specific outcomes, Impact on performance	Application domains, Task-specific requirements
RQ5	LLM researchers, developers, industry stakeholders	Identified challenges, Limitations in existing LLM models	Addressing challenges, Pushing the boundaries in language understanding	Current state of LLM technology, Evolving NLP landscape
RQ6	Developers, policymakers, ethicists	Ethical guidelines, Bias mitigation strategies	Ethical deployment, Reduced biases and concerns	Societal impact, Regulatory frameworks

Table 2 Outcome of the literature database search

Selected databases	Count	Reduced count
IEEE Xplore	28	13
Web of Science	21	03
Google Scholar	147	32
Emerald	20	3
Scopus	32	02
Total	248	53

Table 3 Inclusion and exclusion criteria

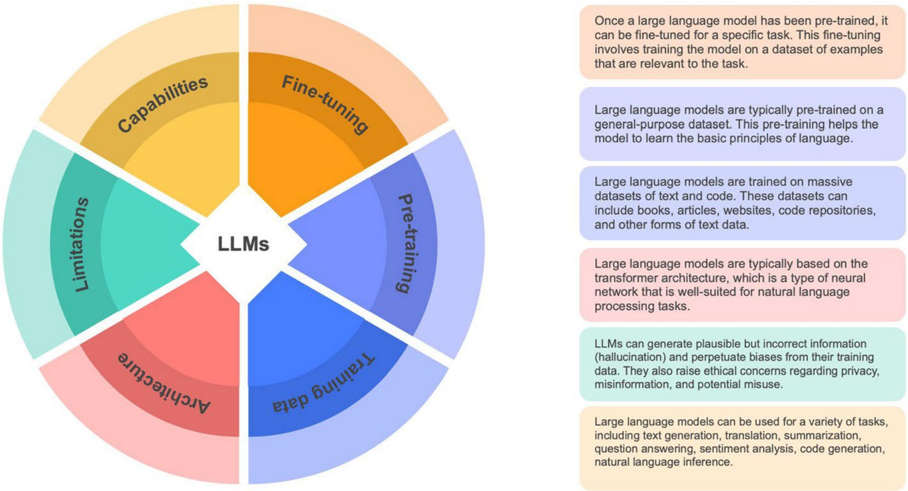
Nos.	Item	Basis of inclusion	Basis of exclusion
1	LMs	NLP tasks	Only NLP tasks
2	Multi-task learning (MTL)	Combination of tasks	Only single task
3	Year	2014–2023	< 2010
4	RQs	Related to at least one RQ	Not related to RQ

Table 4 Criteria for evaluating quality

Metric	Scores
Results presented	1/0
Empirical evidences presented	1/0
Clear objectives & objectives	1/0
Proper references provided	1/0

4 Background

Probability distributions over a sequence of words are scrutinized within the scope of language modeling, a longstanding and foundational pursuit in NLP (Zhang et al. 2022). Language models (LMs) have proven exceptionally beneficial across a spectrum of computational linguistic tasks, including but not limited to speech recognition and text generation. Such tasks derive substantial advantages from the integration of these refined LMs. Figure 2 provides a comprehensive depiction of LLMs. Conventional language models, also referred to as CLMs, are employed to probabilistically predict the occurrence of linguistic sequences. Both CLMs and probabilistic language modeling (PLM) approaches are amenable to training. Presently, data-driven methodologies are ubiquitous across organizations (Wei et al. 2023). PLMs are crafted by training neural network models on extensive datasets sourced from diverse corpora. The PLMs are subsequently adjusted through fine-tuning procedures utilizing datasets and objectives customized to suit their intended applications. The probability of a word sequence within a conventional language model can be approximated through techniques such as n-grams or Hidden Markov Models. The chain rule is one method that can be utilized when calculating the probability: $P\left(w_1, w_2, \ldots, w_n\right) = P\left(w_1\right) \cdot P\left(w_2 \mid w_1\right) \cdot P\left(w_3 \mid w_1, w_2\right) \cdot \ldots P\left(w_n \mid w_1, w_2, \ldots, w_{n-1}\right)$, where $\left(w_1, w_2, \ldots, w_n\right)$ represents sequence of words. Pre-trained LMs, such as GPT or



Once a large language model has been pre-trained, it can be fine-tuned for a specific task. This fine-tuning involves training the model on a dataset of examples that are relevant to the task.

Large language models are typically pre-trained on a general-purpose dataset. This pre-training helps the model to learn the basic principles of language.

Large language models are trained on massive datasets of text and code. These datasets can include books, articles, websites, code repositories, and other forms of text data.

Large language models are typically based on the transformer architecture, which is a type of neural network that is well-suited for natural language processing tasks.

LLMs can generate plausible but incorrect information (hallucination) and perpetuate biases from their training data. They also raise ethical concerns regarding privacy, misinformation, and potential misuse.

Large language models can be used for a variety of tasks, including text generation, translation, summarization, question answering, sentiment analysis, code generation, natural language inference.

Fig. 2 Broad overview of LLMs

BERT, utilize neural networks and fine-tuning for various NLP tasks. In these models, the probability of a word sequence is computed using the Transformer architecture. Given a sequence of words $x_1, x_2, \ldots, x_n$, the probability can be calculated using the following equation:

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid x_1, x_2, \ldots, x_{i-1}) \tag{1}$$

In the context of the given formulation, $P(x_i \mid x_1, x_2, \ldots, x_{i-1})$ denotes the conditional probability of the ith word given the sequence of preceding words up to $i-1$, and x_i denotes the ith word in the input sequence.

4.1 CLMs and PLMs

CLMs and PLMs refer to different types of language models used in NLP. Here are the key differences between the two:

- *Context*
  - CLM: a CLM generates text conditioned on a given input or context. It takes into account the previous words or context to generate the next word or sequence of words. CLMs are often used in tasks like machine translation, chatbots, and text completion.
  - PLM: in order to predict the likelihood of the next word in a sequence, A PLM is trained using a significant amount of text (Fu-Hao et al. 2022) from a corpus. It is able to generate text on its own and does not need to be explicitly conditioned on a particular context in order to do so. PLMs are widely used for various NLP tasks, including text generation, sentiment analysis, named entity recognition, and more.

- *Training*
  - CLM: an average CLM undergoes training through supervised learning, wherein it learns to predict the following word or word sequence based on a given context. During the training process, the model is provided with pairs of input contexts and their corresponding target outputs as the training data (Wei et al. 2023).
  - PLM: a PLM undergoes unsupervised learning on an extensive collection of text, where it gains proficiency in predicting the succeeding word in a sequence by analyzing the preceding words. Typical approaches employed to train PLMs encompass next sentence prediction (NSP) and masked language modeling (MLM).
- *Autoregressive vs. Autoencoding*
  - CLM: CLMs are autoregressive models, meaning they generate text by sequentially predicting each word based on the previous context. Autoregressive models can be slower during generation since they generate words one at a time.
  - PLM: PLMs can be both autoregressive and autoencoding models (Wei et al. 2023). In addition to generating text autoregressively, they can also perform tasks like text classification or named entity recognition by encoding the input text and making predictions based on the learned representations.
- *Fine-tuning*
  - CLM: CLMs are often fine-tuned on specific downstream tasks. The pretrained model is adapted to the target task by further training on task-specific data. This fine-tuning helps the model specialize in the desired task and improve performance.
  - PLM: PLMs can also be fine-tuned for specific tasks, similar to CLMs. The pre-trained PLM serves as a feature extractor, and the additional task-specific layers are trained on the labeled data (Fu-Hao et al. 2022). As shown in Fig. 3, a chain of Transformers serves as the backbone of the encoder, while a language model that has already been trained is used for decoding. To enhance the input acoustic feature frames and encompass short-term temporal variances, the method initially incorporates two convolutional neural network (CNN) layers employing 2D filters and subsampling for each speech signal. The stride of each CNN layer is set to 2, reducing the quantity of acoustic representations to one-fourth. To signify the long-term dependencies across all acoustic statistics, six Transformers are sequentially arranged, with position embeddings employed to encode the absolute position of each acoustic representation within an utterance. The encoder and decoder are surrounded by green and red dashed lines.

Overall, CLMs and PLMs serve different purposes in NLP. CLMs are conditioned on specific input contexts and generate text accordingly, while PLMs are pretrained models capable of generating text without explicit conditioning. Both types of models have their own strengths and are widely used in various NLP applications.

With the rise of neural networks and deep learning, PLMs have garnered significant attention. Examples of such models include OpenAI's GPT series and BERT, which are trained on large textual datasets using unsupervised learning methods (Devlin et al. 2018; Liu et al. 2019). During training, these models predict missing words in sentences or estimate the probability of a word given its context within a broader text (Yang et al. 2019; Lan et al. 2019). PLMs possess the capacity to capture abundant semantic and syntactic information from the training data and can be fine-tuned for specific NLP tasks by utilizing task-specific datasets and objectives. The process of fine-tuning customizes the pre-existing

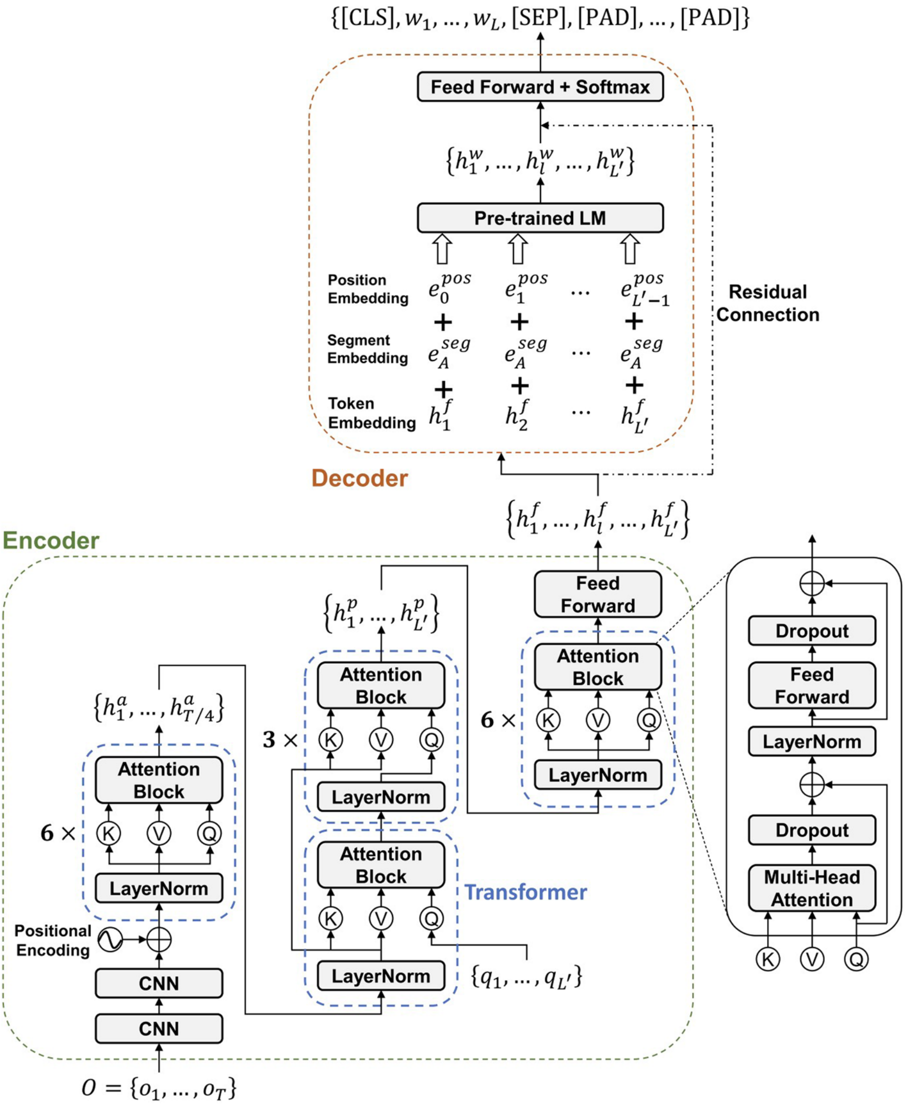

{[CLS], w_1, ..., w_L, [SEP], [PAD], ..., [PAD]}
Feed Forward + Softmax
$\{h_1^w, \ldots, h_l^w, \ldots, h_{L'}^w\}$
Pre-trained LM
Position Embedding
e_0^{pos} e_1^{pos} ... $e_{L'-1}^{pos}$
Segment Embedding
e_A^{seg} e_A^{seg} ... e_A^{seg}
Token Embedding
h_1^f h_2^f ... $h_{L'}^f$
Residual Connection
Decoder
$\{h_1^f, \ldots, h_l^f, \ldots, h_{L'}^f\}$
Encoder
Feed Forward
Attention Block
6 ×
K V Q
LayerNorm
$\{h_1^p, \ldots, h_{L'}^p\}$
3 ×
Transformer
$\{q_1, \ldots, q_{L'}\}$
$\{h_1^a, \ldots, h_{T/4}^a\}$
Positional Encoding
CNN
CNN
$O = \{o_1, \ldots, o_T\}$
Dropout
Feed Forward
LayerNorm
Dropout
Multi-Head Attention

Fig. 3 The acoustic representation encoder and text generation decoder make up the network architecture of the proposed non-autoregressive ASR framework (Fu-Hao et al. 2022). (Color figure online)

model for a specific use case, such as machine translation, sentiment analysis, or question answering. By leveraging pre-existing knowledge and customizing it for specific tasks, organizations can capitalize on data-driven approaches across various NLP applications.

4.2 Architectures

In the subsequent section, language models, also termed Transformer-based language models are examined, and synopsis of each is provided. These language models, employing a specialized form of deep neural network architecture known as the Transformer, aim to predict upcoming words in a text or words masked during the training process. Since 2018, the fundamental structure of the Transformer language model has scarcely changed (Radford et al. 2018; Devlin et al. 2018). An advanced architecture for sharing information about weighted representations amongst neurons is the Transformer (Vaswani et al. 2017). It utilizes neither recurrent nor convolutional architectures, relying solely on attention processes. To learn the most relevant information from incoming data, the Transformer's attention mechanism assigns weights to each encoded representation. To determine the result of the attention operation, one computes the weighted sum of the values to yield the outcome. The compatibility function between the query and the corresponding key is utilized to determine how much weight each factor should be given Vaswani et al. (2017). The evolution of LMs has led to the proliferation of various attention mechanisms (Guo et al. 2022). For example, self-attention is developed in NLP to establish associations between nodes in a sequence to generate a representation of that sequence. The Transformer model utilizes a mask matrix to establish the visibility of words to each other, facilitating an attention mechanism grounded in self-attention.

The initial step involves parsing a string of text into a sequence of tokens, some of which may comprise multiple smaller tokens due to limitations in available vocabulary. Subsequently, each token undergoes an "embedding" process, wherein it is allocated a fixed vector. These vectors are acquired during the pre-training phase (Vaswani et al. 2017). A series of Transformer layers, typically ranging from 10 to 100 layers, is utilized for processing the embeddings. These layers consist of individual feedforward networks, layer normalizations, and self-attention networks at the token level. The self-attention network, a pivotal and innovative component of the Transformer layers, generates "value," "query," and "key" vectors for each token, employing projections. Through this mechanism, token embeddings are amalgamated to produce a "contextualized" representation of each token, thereby completing the transformation process. Essentially, this architecture converts each input token into a distribution over forthcoming tokens based on their probabilities. Typically, the number of parameters in a language model ranges between 100 million and 500 billion (Devlin et al. 2018; Brown et al. 2020; Lieber et al. 2021; Smith et al. 2022); autoregressive models tend to possess more parameters than masked models.

The Transformer architecture, as originally formulated, does not inherently encode positional information of tokens within input sequences, despite its potential utility in capturing word order. Consequently, Transformer-based language models integrate various methods to incorporate positional information (Wang et al. 2021; Dufter et al. 2022). These methods include augmenting token embeddings with absolute position embeddings (Vaswani et al. 2017; Radford et al. 2018; Brown et al. 2020; Zhang et al. 2022), utilizing relative position embeddings or biases (Shaw et al. 2018; Dai et al. 2019; Raffel et al. 2020), or employing rotary position embeddings (Su et al. 2021; Chowdhery et al. 2022). Studies suggest that models employing relative position approaches may exhibit improved performance in extrapolating to longer sequences compared to those utilizing absolute position methods (Press et al. 2021). Input sequences for language models typically range between 500 and 2000 tokens in length prior to pre-training.

4.3 Learning methodologies

Studies indicate that deep learning algorithms within the field of computer vision (CV), display superior efficacy in contrast to traditional learning algorithms across a broad spectrum of tasks. These tasks include but are not limited to classification, identification, detection, and segmentation, as well as more specialized tasks such as matching, tracking, and sequence prediction. The disparity in performance between deep learning and traditional models is considerable across these tasks. Furthermore, both natural language processing (NLP) and graph learning (GL) employ analogous learning methodologies (Samant et al. 2022).

- *Supervised learning*

  Substituting every instance of Y found within the training dataset with values denoted as $\{(a_i, b_i)\}_{i=1}^{n}$ facilitates a faithful representation of the original dataset. As a result, the variable Y can be represented as $\{(a_i, b_i)\}_{i=1}^{n}$, where a_i signifies the ith instance in the training set and b_i signifies its associated label. The overarching objective of the network is to minimize the subsequent objective function (Zhou et al. 2023) while acquiring proficiency in learning a function $f(x;\theta)$:

$$\arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} L\big(f(a_i;\theta), b_i\big) + \lambda\Omega(\theta) \tag{2}$$

  where the value of L along with the value of Ω is the fixed loss function value and an additional regularization term.

- *Semi-supervised learning*

  Assuming that, in addition to the dataset annotated by humans, there is access to another, unlabeled dataset $Z = \{z_i\}_{i=1}^{m}$. Utilizing both datasets collectively, a learning approach is devised to acquire an optimal network, as outlined in Zhou et al. (2023):

$$\arg\min_{\theta} \left(\frac{1}{n} \sum_{i=1}^{n} L\big(f(a_i;\theta), b_i\big) + \frac{1}{m} \sum_{i=1}^{m} L_0\big(f_0(z_i;\theta_0), R(z_i, X)\big) + \lambda\Omega(\theta) \right)$$

  Within this framework, a relationship function denoted as $R$is defined to outline the objectives pertaining to the untagged data. These pseudo-labels are merged into the overall training regimen. The dataset Z retains the primary data, and an encoder, denoted as f_0, is employed to generate a new depiction of this data. Essentially, unsupervised learning and self-supervised learning (SSL) facilitate learning from inherent data characteristics without the presence of labeled data during training. This is accomplished by examining internal distances or predefined pretext tasks.

- *Reinforcement learning*

  As soon as an event of type t occurs, the agent is given a state p_t from a state space P of its choosing. In the next step, an action q_t is chosen from an available set of actions Q using a policy $\pi_\theta(q_t|p_t)$, where π_θ that maps actions to states according to some parameters θ. The agent is then rewarded with a scalar $s_t = s(p_t, q_t)$ and progresses to state p_{t+1} based on the dynamics of the surrounding environment (here, the notation $r(p, q)$ represents the reward function). This procedure is repeated after each episode until the agent reaches an endpoint. As soon as one episode concludes, the RL agent will restart and begin the next. Total rewards are discounted by a factor of $\gamma \in (0, 1]$, defined as $S_t = S(p_t, q_t) = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ resulting in a net reward for each condition. The agent's goal is to achieve the highest possible long-term expected return from each state as shown below (Zhou et al. 2023):

$$\max_{\theta} \mathbb{E}_{\pi_\theta}[S_t | p_t, q_t = \pi_\theta(p_t)] \tag{3}$$

4.4 Linguistic units

In the domain of English language modeling, tokenization denotes the procedure of dissecting a textual sequence into diminutive units termed as tokens, serving as the fundamental components for language models. The choice of tokenization methodology hinges upon the specific language and model utilized. Presented below are numerous prevalent techniques for tokenization utilized in English language modeling, contingent upon diverse unit dimensions:

- *Character-level tokenization*: this method treats every single character within the text sequence as an individual token. For instance, the sentence “Hello, world!” would be tokenized into [‘H’, ‘e’, ‘l’, ‘l’, ‘o’, ‘,’, ‘ ’, ‘w’, ‘o’, ‘r’, ‘l’, ‘d’, ‘!’]. Character-level tokenization is advantageous when dealing with languages lacking clear word boundaries or for specific tasks like character-level language modeling (Sutskever et al. 2011; Al-Rfou et al. 2019; Xue et al. 2022). In mT5 (Xue et al. 2020), the text is segmented into SentencePiece tokens, and segments of approximately 3 tokens are masked (highlighted in red). Both the encoder and decoder transformer stacks have identical depths. In contrast, ByT5 (Xue et al. 2022) processes the text as UTF-8 bytes, with spans of approximately 20 bytes being masked. Additionally, the encoder in ByT5 is three times deeper than the decoder.
- *Word-level tokenization:* this technique divides the textual sequence into discrete units comprising single words or word-like entities (Sennrich et al. 2015). For instance, the phrase “Hello, world!” undergoes tokenization resulting in [‘Hello’, ‘,’, ‘world’, ‘!’]. Utilizing word-level tokenization is a prevalent practice as it affords a straightforward depiction of text, enabling the model to apprehend the semantic essence of individual words and their interconnections within a sentence.
- *Subword-level tokenization:* this approach involves the segmentation of words into smaller subword units, which can aid in managing out-of-vocabulary terms or accommodating languages with complex morphology (Mikolov et al. 2012). Subword tokenization techniques such as Byte-Pair Encoding (BPE) (Gage 1994) or Unigram Language Model (ULM) (Kudo 2018) are commonly employed to construct a vocabulary of subword units based on the training corpus. For instance, the term “unhappiness” could be segmented into [‘un’, ‘happiness’] or [‘un’, ‘h’, ‘ap’, ‘p’, ‘i’, ‘ness’]. Tokenization at the subword level enables the model to handle novel words by leveraging the subword units found in the training data.
- *Phrase-level tokenization:* in certain scenarios, sequences of words or multi-word expressions have the potential to be regarded as tokens (Suhm 1994; Saon and Padmanabhan 2001). This method entails representing the semantic content of frequently encountered phrases as a singular entity, as opposed to dissecting them into separate words (Levit et al. 2014). For instance, the expression “New York City” could be tokenized as [‘New York City’]. The practice of tokenization at the phrase level proves

advantageous, especially when particular phrases carry substantial semantic or contextual relevance within the given task.

The choice of tokenization method depends on the specific requirements of the language modeling task and the characteristics of the language under consideration. Various tokenization strategies possess unique benefits and constraints, prompting researchers to explore different methods to identify the most appropriate one for their particular application (Table 5).

4.5 Training dataset

The precise forecasting of performance enhancement in generative models between 2019 and 2022 has precipitated a notable expansion in the scale of these models within this timeframe. With increased model size, given adequate data and computational resources, proficiency in word prediction within textual contexts, as demonstrated by the training dataset, is enhanced (Kaplan et al. 2020; Ganguli et al. 2022; Hoffmann et al. 2022). Consequently, a pre-trained LLM comprehends a broader array of contexts, yielding higher-quality, more nuanced, and lengthier texts while retaining greater contextual coherence with preceding text passages. The performance of predictive base models is inherently intriguing, yet a noteworthy transition occurs as models undergo size augmentation. Noteworthy is the capacity of LLMs, equipped with between 10 and 100 billion parameters, to undertake specialized tasks such as code generation, translation, and human behavior prediction, often surpassing or matching the proficiency of specialized models. Table 6 displays the analysis of several prominent and initial PLMs. Anticipating the emergence of such capabilities has posed challenges, and the potential additional capabilities of larger models remain uncertain (Ganguli et al. 2022) (Table 7).

However, as LLMs increase in size, they reveal not only new capabilities but also new failure mechanisms. The development of biases related to sex, gender, race, and religion in LLMs coincides with their learning in programming and chess (Ganguli et al. 2022).

5 Word embeddings

In the domain of LLMs, the term "word embeddings" refers to the representation of words as condensed, lower-dimensional vectors within a continuous vector space. These embeddings encapsulate both semantic and syntactic associations among words, derived from their co-occurrence patterns within a specified text corpus (Petukhova et al. 2024). In the context of word embeddings, the term "lower-dimensional" is used to compare the vector representations of words with the original high-dimensional space in which the words exist. These original high-dimensional spaces typically represent the entire vocabulary of words in a language, where each word is represented by a one-hot encoded vector of size equal to the vocabulary size. The word embeddings acquired by extensive language models encapsulate comprehensive contextual details and encode semantic associations among words. They facilitate the comprehension of word meanings by the models and facilitate their utilization in various natural language processing endeavors. In this section, various prominent methodologies for producing word embeddings are examined and a concise summary of the typically utilized approaches is provided.

Table 5 An overview of recently developed LLMs and their specifications

Model	Year	Parameters	Layers	Dataset	Training dataset size	Architecture	Key features
GPT (Radford et al. 2018)	2018	117 M	12	1B-WLMB	11 GB	Transformer Decoder	Pre-training (generative) a language model to generate predictions from an unlabeled text corpus
BERT-base (Devlin et al. 2018)	2018	110 M	12	BookCorpus + English Wikipedia	1.2 GB	Transformer Encoder	Joint conditioning on left and right context across all layers to generate deep bidirectional representations from unlabeled text
Transformer-XL (Dai et al. 2019)	2019	257 M	18	Multiple datasets	n/a	Transformer	A new positional encoding scheme and a recurrence mechanism at the segment level
GPT-2-Large (Radford et al. 2019)	2019	774 M	36	WebText + BookCorpus	40 GB	Transformer Decoder	Improvements in performance are logarithmic in nature, and the model's capacity is crucial to the success of zero-shot task transfer
DeBERTa (base) (He et al. 2021)	2020	140 M	12	Wikipedia + BookCorpus	~1.18 GB	Transformer Encoder	Attention mechanism encodes each word's content and position with two vectors, then, an enhanced mask decoder predicts masked tokens using absolute positions
MARGE (Lewis et al. 2020)	2020	960B	12	MLSum, PAWS-X, MLQA	1.2, 3.5, 1.7 GB	Seq2Seq	The target text is replicated by accessing a repository of similar texts written in different languages
XLM-E (Chi et al. 2022)	2021	279 M	12	Wikipedia + CommonCrawl + BookCorpus	1.3 TB	Transformer	Use of the multilingual replaced token (MLT) detection and translation replaced token (TRT) detection
Gopher (Rae et al. 2022)	2021	280B	137	Wikipedia + CommonCrawl + BookCorpus + CodeSearch	10.5 TB	Transformer	Checking facts, comprehension reading and the toxic language identification are aided the most from scale

Table 5 (continued)

Model	Year	Parameters	Layers	Dataset	Training dataset size	Architecture	Key features
LaMDA (Thoppilan et al. 2022)	2022	137B	137	Wikipedia + CommonCrawl + BookCorpus + CodeSearch + Stack Overflow + Reddit	10.5 TB	Transformer	Scaling up model alone boosts quality, but has less of an effect on security and credibility
GPT-NeoX-20B (Black et al. 2022)	2022	20 B	137	Wikipedia + CommonCrawl + BookCorpus + CodeSearch + Stack Overflow + Reddit + Pile	n/a	Transformer Decoder	A highly effective few-shot reasoner that outperforms GPT-3 and Fair-Seq models of comparable size in five-shot evaluations
PaLM (Chowdhery et al. 2022)	2022	540 B	137	Wikipedia + CommonCrawl + BookCorpus + CodeSearch + Stack Overflow + Pile + Infiniset	n/a	Transformer	Discontinuous gains from model scale were observed in a large fraction of BIG-bench tasks
OPT-66 B Zhang et al. (2022))	2022	175 B	64	Wikipedia + CommonCrawl + BookCorpus + CodeSearch + Stack Overflow + Pile + Infiniset	n/a	Transformer Decoder	OPT-175B is similar to GPT-3, but its carbon footprint is only one-seventh as large

Table 6 Analysis of select initial pre-trained language models

PLM	Vocabulary size	Tokenizer type	Pretraining data size
GPT-3 (OpenAI)	175 billion	Byte pair encoding	~ 570 GB
BERT (Google)	30,000 (WordPiece)	WordPiece	~ 3.3 TB
T5 (Google)	800 million	SentencePiece	~ 7 TB
XLNet (Google)	32,000 (SentencePiece)	SentencePiece	~ 126 GB
RoBERTa (Facebook)	50,265	Byte pair encoding	~ 160 GB
ELECTRA (Google)	50,000	WordPiece	~ 1.3 TB
GPT-2 (OpenAI)	1.5 billion	Byte pair encoding	~ 40 GB

5.1 Latent semantic analysis (LSA)

The objective of LSA is to extract and depict word usage within context (Asudani et al. 2023). This facilitates the identification of clusters of words with akin meanings. LSA, employing a vector space methodology, enables the extraction of word-word, passage-passage, and passage-passage relationships (Shaik et al. 2022). These relationships, alongside human cognitive processes and semantic representations, exhibit significant interconnections. The utilization of LSA substantially enhances our capacity to extract and discern the semantic associations that manifest in the thoughts of speakers or writers during communication. Furthermore, LSA can be utilized to approximate human judgment, aiding in the anticipation of semantic connections among different textual segments, and providing computational estimations of semantic similarities between words (Singh et al. 2022; Zhang et al. 2022; Al-Hashedi et al. 2022).

LSA entails a two-phase, fully automated mathematical-statistical process. The initial stage involves constructing a matrix wherein each row corresponds to unique words within the input text, while sentences, sections, and other textual entities (e.g., paragraphs and documents) are allocated their individual columns. Initially, LSA generates a matrix representation of the input text denoted as a term-by-sentence matrix (A). For the ith phrase, every (A_i)th column vector in the matrix constitutes a term-frequency vector accompanied by respective weights. Given s sentences and t distinct words or terms within the input text, matrix A manifests as a $t \times s$ matrix ($t \gg s$). Word and phrase frequencies within a specific sentence are logged in each cell of the matrix. Various methods such as the count of occurrences, binary representation of occurrences, TF-IDF, Root Type, Log Entropy, and Modified TF-IDF can be employed to determine the values of matrix cells.

Singular Value Decomposition (SVD) is then applied to matrix A in the second stage of LSA. Matrix A is triangulated into matrices V, Σ, and U after being subjected to SVD (Ramezani et al. 2023).

$$A_{t \times s} = U_{t \times c} \Sigma_{c \times c} V^{T}_{c \times s} \tag{4}$$

In this case, U is a $t \times c$ column-orthogonal matrix with left singular values in its columns. If A has eigenvalues, then the principal diagonal elements of $\Sigma = \text{diag}(\delta_1, \delta_2, \ldots, \delta_c)$ will also be eigenvalues of $c \times c$ as a whole. On the major diagonal, the eigenvalues have been arranged from least to greatest (Ramezani et al. 2023). The columns of the V^T matrix, which are the right singular values, are those of the orthogonal $c \times s$ matrix. The following relation is given for the matrix Σ if and only if the condition $\text{rank}(A) = r$ holds: $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_r \geq \sigma_{r+1} = \cdots = \sigma_s = 0$

5.2 Word2Vec

Word embeddings are created using a model proposed by Mikolov et al. (2013), developed by Google. Word2vec differentiates itself from LSA and multi-model learning (MML) by being a predictive rather than a statistical model. The underlying linguistic framework of Word2vec is based on a feedforward neural network model (Bengio et al. 2000). Word2vec employs both the Skip-Gram and continuous bag of words (CBOW) models, which utilize neural networks. The Skip-Gram model operates by predicting the surrounding words of a target word, whereas the CBOW model predicts the current word by aggregating the word vectors of its neighboring words. By leveraging a contextual window, Word2Vec is capable of unsupervised learning to determine semantic meaning and similarity between words (Zhao et al. 2022; Subba and Kumari 2022; Oubenali et al. 2022). Terms with similar meanings (such as “king” and “queen”) typically cluster together within this semantic space. CBOW models are more efficient than Skip-Gram models since they treat the entire context as a single entity, rather than generating multiple training pairs for each word in the context. However, the Skip-Gram model performs better at identifying rare words due to its superior context management.

Word2Vec has experienced significant growth in NLP in recent years (Mikolov et al. 2013). The primary function of the trained model is to provide weights for the word embedding technique, which relies on a shallow neural network. These embedding vectors capture the most salient word connections in the training set. The modeling capabilities of Word2Vec extend beyond the NLP domain. During training, Word2Vec models require the specification of the target vector length, denoted by N. Another critical parameter is the window length W, representing the width of the sliding window used to extract training samples from the data. Word2Vec embeddings possess algebraic features. For instance, consider an advanced Word2Vec model trained on English text. Let $V(w_i)$ be the Word2Vec embedding of word w_i, where w_0 = “king”, w_1 = “man”, w_2 = “woman”, w_3 = “queen”. According to Mikolov et al. (2013), if “closeness” is measured by cosine similarity, the vector $V(w_3)$ is most similar to the vector $V(w_0) - V(w_1) + V(w_2)$. The results suggest that the embeddings generated by Word2Vec effectively represent key elements of linguistic semantics within the domain of NLP.

5.3 Global vectors (glove)

GloVe, short for Global Vectors for Word Representation, is an unsupervised learning algorithm designed to generate word embeddings. It creates a vector space representation of words based on their co-occurrence statistics within a large text corpus (Pimpalkar 2022). The GloVe algorithm utilizes word co-occurrence data or global statistics to infer semantic relationships between words in the corpus, as opposed to word2vec’s dependence on local context windows for deriving these associations (Badri et al. 2022; Gan et al. 2022; Curto et al. 2022). To identify word co-occurrences, GloVe employs a global matrix factorization technique (Pennington et al. 2014). While word2vec is based on a feedforward neural network model, making it a “neural word embeddings” technique, GloVe is based on a log-bilinear function, classifying it as a “count-based” model. By analyzing the frequency of word pair co-occurrences in a corpus, GloVe captures their relationships. The ratio of the probabilities of two words co-occurring can encode meaning and assist in addressing the word-analogy problem.

The initial stage of GloVe involves the creation of a co-occurrence matrix. Consider a vocabulary of size V, and each word is represented by its index in the vocabulary. The co-occurrence matrix, denoted by X, is a $V \times V$ matrix, where each element X_{ij} represents the number of times word i and word j co-occur within a specific context window. The objective of GloVe is to acquire word vectors that represent both the semantic and syntactic associations among words. It does so by defining a word vector for each word, denoted by w, and a context vector, denoted by c. The word vectors and context vectors are both of size d, representing the dimensionality of the word embeddings (Pennington et al. 2014).

GloVe defines an objective function that measures the similarity between word vectors and context vectors:

$$J = \sum_{i,j} \left[w_i^T \cdot c_j + b_i + b_j - \log(X_{ij}) \right]^2 \tag{5}$$

where w_i and c_j are the word vector and context vector for words i and j, respectively, and b_i and b_j are the corresponding bias terms. The objective function J is optimized using gradient descent or other optimization algorithms. The goal is to minimize the difference between the dot product of word and context vectors, the biases, and the logarithm of the co-occurrence counts. The GloVe algorithm introduces a parameter α to control the importance of each co-occurrence pair. The co-occurrence counts are raised to the power of α, resulting in the following equation (Pennington et al. 2014):

$$J = \sum_{i,j} f(X_{ij}) \left[w_i^T \cdot c_j + b_i + b_j - \log(X_{ij}) \right]^2 \tag{6}$$

where $f(X_{ij}) = \min\left(1, \left(\frac{X_{ij}}{x_{\max}}\right)^{\alpha}\right)$ is a weighting function. $x_{\max}$ represents the maximum co-occurrence count in the matrix. By optimizing this objective function, GloVe learns the word vectors and context vectors that capture meaningful relationships between words.

5.4 FastText

This technique involves learning large-scale word embeddings (Bojanowski et al. 2017; Joulin et al. 2016), representing an advancement from Word2Vec. FastText considers each word as a composite of character n-grams (e.g., “unbalanced” = “un” + “balance” + “ed”), rather than as single units, with the objective of learning vector representations by leveraging the character and morphological structure of words. Consequently, words can be represented by averaging the embeddings of their n-grams. Although the initial computational cost is higher, fastText’s efficient representation of words through sub-word components allows it to estimate “out of vocabulary” (OOV) and rare words, as their character-based n-grams are likely shared with other words in the training data.

The Facebook AI research team developed the FastText library, which is a compilation of word representations. It includes 2,000,000 frequent crawl words, each represented by 300 dimensions, resulting in 600,000,000 word-vectors. Besides single words, it incorporates hand-crafted n-grams as features. Due to its simple design, text classification can be performed efficiently and accurately (Qiao et al. 2018). Word embedding methods have been widely applied to various text categorization problems. Pre-trained word embeddings can predict word contexts in an unsupervised manner, assuming that words in close proximity within a sentence have similar meanings (Badri et al. 2022; Kowsher et al. 2022). FastText embeddings utilize morphological cues to accurately represent vectors, aiding in the identification of problematic words. This capability also enhances its generalizability (Didi et al. 2022). To improve the handling of unfamiliar words, FastText word embedding employs n-grams to construct vectors.

5.5 Contextualized word embeddings (CoVe)

The objective of CoVe (McCann et al. 2017) is to enhance word representation by training an encoder and subsequently adapting it for a different task. By assigning zeros to represent unknown words, this approach also encounters the OOV issue (McCann et al. 2017; Mars 2022), as depicted in Fig. 4. Bilateral attention deficits, a maxout network calculates a distribution over classes using pooled features from multiple representations, a biLSTM incorporates conditional information from both representations, and the two methods are interdependent.

The BiLSTM accepts a word sequence as input and produces a series of hidden states representing the contextual details of each word. Taking the input sequence of words as $X = [x_1, x_2, \dots, x_T]$, where T is the length of the sequence. Each word x_t is associated with a word embedding vector e_t. The hidden states of the BiLSTM at each time step are denoted as $H = [h_1, h_2, \dots, h_T]$. The forward pass of the BiLSTM involves computing the hidden states for each time step t using the input sequence X. The forward hidden states h_t are computed as follows:

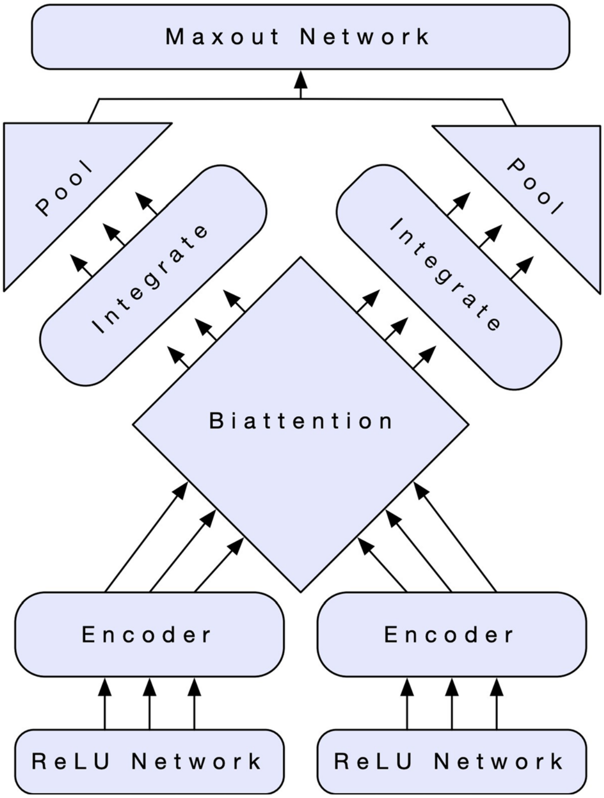


Fig. 4 Each input sequence is represented in a unique way for the given task using a feed-forward network with ReLU activation and a biLSTM encoder (McCann et al. 2017)

$$h_t = \text{LSTM\_forward}(e_t, h_{t-1}) \tag{7}$$

where LSTM_forward is the forward LSTM cell function and h_{t-1} is the previous hidden state. Similarly, the backward pass of the BiLSTM involves computing the backward hidden states g_t using the input sequence X. The backward hidden states g_t are computed as follows:

$$g_t = \text{LSTM\_backward}(e_t, g_{t+1}) \tag{8}$$

The LSTM_backward function represents the backward LSTM cell, and g_{t+1} denotes the subsequent hidden state. The combined forward and backward hidden states: $c_t = [h_t, g_t]$ are concatenated to derive the ultimate contextualized word representation c_t. The hidden states h_t and g_t capture the contextual information of the word x_t, considering both preceding and succeeding words. The encoder is trained by optimizing a task-specific objective function, such as language modeling or machine translation. Once the encoder is trained, it can be repurposed for a different task by using the contextualized word representations c_t as inputs to downstream models or classifiers (Mars 2022; Andrabi and Wahid 2022; Ghanem and Erbay 2023). The training procedure entails the optimization of an objective function specific to the task, yielding an encoder capable of producing contextualized word embeddings suitable for diverse downstream tasks.

6 Deep learning for LLM tasks

Text classification (TC) is a fundamental sub-task underpinning all natural language understanding (NLU) tasks. Questions and answers from customer interactions exemplify text data originating from various sources. While text provides a robust data foundation, its lack of organization complicates the extraction of meaningful insights, making the process challenging and time-consuming. TC can be performed using either human or machine labeling. The increasing availability of data in text form across various applications underscores the utility of automated text categorization. Automatic text classification typically falls into two categories: rule-based or artificial intelligence-based methods. Rule-based approaches categorize text based on predefined criteria and require extensive domain expertise. AI-based methods, on the other hand, are trained on labeled text samples to classify new texts. Machine learning (ML) algorithms learn the relationship between the text and its labels. Traditional ML-based models often follow a phased approach. Generally, NLU is employed for tasks requiring reading, understanding, and interpretation. The first step involves manually extracting features from a document, and the second step involves fitting these features into a classifier to generate a prediction. Relying on manually extracted features necessitates complex feature analysis to achieve reasonable performance, which is a limitation of this phased approach.

OpenAI has harnessed Google's innovative Transformer neural network architecture (Vaswani et al. 2017) for embedding model creation since 2018. The attention-based Transformer significantly enhances the accuracy of TPU-based model training on a large scale. GPT (Radford et al. 2018), an influential framework developed utilizing Transformers, is currently prevalent for text generation tasks, as depicted in Fig. 5. In 2018, BERT (Devlin et al. 2018), an advancement over the bidirectional transformer, was introduced by Google. OpenAI's latest GPT-3 model (Brown et al. 2020) continues in the same trajectory, employing larger models trained on expanded datasets. This section highlights

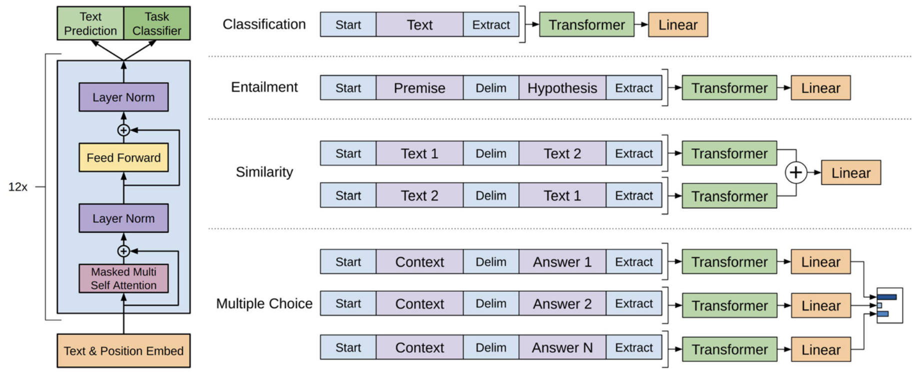


Fig. 5 (Left) The framework and training targets for the Transformer (Radford et al. 2018). (Right) Adjustments to the input data for tuning performance on specific tasks

notable advancements in NLP and natural language understanding (NLU), demonstrating the application of various deep learning models for diverse language comprehension tasks within LLM.

6.1 Feed forward neural networks

In the domain of NLP (Radford et al. 2018; Devlin et al. 2018; Liu et al. 2019; Radford et al. 2019; Brown et al. 2020), the efficacy and adaptability of large-scale pretrained language models are noteworthy. It has been evidenced that augmenting the model size (scaling) serves as a dependable strategy for enhancing generalization, thereby facilitating additional functionalities, without encountering performance plateaus (Kaplan et al. 2020; Zhang et al. 2022; Chowdhery et al. 2022; Hoffmann et al. 2022; Wei et al. 2022). Nonetheless, there remains a necessity for more resource-efficient methodologies concerning the training and inference of LLMs, given the considerable computational resources requisite for the development of larger language models.

In the context of a transformer layer, the Feed-Forward Network (FFN) block receives an input vector x from the self-attention block. The FFN block consists of a series of operations that transform the input vector to produce an output vector y. Mathematically, the FFN block can be represented as follows (Liu et al. 2023):

$$y = \mathrm{FFN}(x) = f(x \cdot K^{\top}) \cdot V = m \cdot V \tag{9}$$

value dm, is the total no. of memory cells and the value d is the total no. of dimensions in the input vector x, K is a matrix of shape $dm \times d$ and x is the input vector. The matrix K acts as a set of learnable parameters that are multiplied element-wise with the input vector x. The resulting intermediate vector is then passed through a non-linear function f to obtain the hidden states m, which is a vector of shape dm. The matrix V, with shape $dm \times d$, is another set of learnable parameters. The hidden states m are multiplied with V, resulting in a d-dimensional output vector y. Alternatively, the FFN block can be interpreted as a neural memory. In this view, it consists of dm key-value pairs, which form the key table and the value table, respectively. Each key k_i is a d-dimensional vector, and the values are represented by the matrix V. The memory operation can be represented as follows (Liu et al. 2023):

$$y = \sum_{i=0}^{m-1} m_i \cdot v_i \tag{10}$$

In this equation, the query input x is multiplied with each key k_i, resulting in a memory coefficient m_i for the i-th memory cell. The non-linear function f is applied to each dot product $x \cdot k_i$ to obtain the memory coefficient. The values v_i are the corresponding values from the value table V. Finally, the output vector y is computed as the sum of the values v_i weighted by their respective memory coefficients m_i. Both views, the FFN as a multi-layer perceptron [Eq. (9)] and as a neural memory [Eq. (10)], describe the operations performed in the FFN block of a transformer layer, with the only difference being in how the key-value pairs are represented and utilized. In 2000, authors in Bengio et al. (2000) introduced the first model for analyzing natural language. The training process employs a dataset consisting of 14 million words, and the model is constructed upon a feedforward neural network architecture. Conversely, models that rely on premature embeddings demonstrate inferior performance compared to models that utilize manually acquired features (Borgeaud et al. 2022; Schwartz and Dodge 2020; Tay et al. 2022). Sparse scaling, which increases the amount of parameters while maintaining the same training and inference cost (in FLOPs), is a promising direction. Scaling up the feed-forward network (FFN) of a transformer with sparsely activated parameters has been the subject of recent research, yielding a scaled and sparse FFN (S-FFN). Two primary methods have been developed to accomplish S-FFN. S FFN is viewed as a form of neural memory (Sukhbaatar et al. 2015), and like a sparse memory, it only activates a subset of memory cells upon retrieval (Lample et al. 2019). The other method uses a Mixture-of-Expert Network (MoE) (Lepikhin et al. 2020; Fedus et al. 2022; Roller et al. 2021; Gururangan et al. 2021) that uses multiple, smaller FFN modules (called "experts") and activates a subset of these experts based on the input.

6.2 Recurrent neural networks (RNN)

Models can undergo training on extensive textual datasets, subsequently utilizing the acquired knowledge for subsequent tasks through transfer learning (Mikolov et al. 2013). Before the introduction of the transformer architecture for transfer learning, unidirectional language models were commonly utilized despite their inherent limitations. These limitations encompassed the utilization of one-way RNN architecture and a constrained context vector size. Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al. 2018), can augment the performance of subsequent tasks by addressing these deficiencies.

The LSTM model extends the capabilities of the RNN architecture. While training a basic RNN, the issue of vanishing gradients has been acknowledged. The LSTM model surpasses the standard RNN due to its enhanced memory mechanism. Specifically, the LSTM approach outlined in Hope et al. (2017) effectively reduces dimensionality while achieving notable performance in accurately classifying opinions, owing to its memory function. To employ machine learning for sentiment analysis, simplifying input functions is crucial. For training RNNs on restaurant critics' opinions, researchers in Tarasov (2015) utilized a long short-term memory model. Various methods were employed to evaluate the collected data, including simple recurrent neural networks, logistic regression, bidirectional RNNs, and bidirectional long short-term memory networks. Among these, the best performance was observed with a deep bidirectional LSTM featuring multiple hidden layers. A vector representation of each word can be regarded as a system training parameter to facilitate the analysis of the model's text input. For tasks such as identifying romantic phrases in movie reviews and assessing somatizing sentence pairs, the LSTM model was favored by researchers in Tai et al. (2015). Additionally, in Socher et al. (2013), the Treebank sentiment and recursive neural tensor networks were introduced for emotion recognition tasks. Application of recursive neural tensor networks to the positive/negative categorization of individual sentences resulted in a performance improvement from 80 to 85%.

6.3 Convolutional neural networks (CNN)

In contrast to conventional neural networks, CNNs incorporate neurons with adjustable weights and biases, enhancing their appeal. Each neuron processes multiple inputs via a dot product operation, optionally followed by a nonlinearity function. Consequently, CNNs form a feed-forward architecture that continually evolves (LeCun et al. 2010). Convolution primarily aims to identify pertinent features within datasets that exhibit only local connectivity. The activation function, crucial for learning abstract concepts and introducing nonlinearity into the feature space, receives the convolutional kernels' outputs. The presence of unique activation functions for each neuron due to nonlinearity facilitates learning significant distinctions between images. Furthermore, subsampling typically succeeds the nonlinear activation function's output, imparting resistance to geometric variations in the input and simplifying output summarization.

The CNN has found applications in NLP, such as language modeling and analysis, despite RNNs being deemed more appropriate for these purposes (Bhatt et al. 2021). Since the inception of CNN as a novel representation learning technique, there has been a shift in the methodology of sentence modeling or structuring in language. Sentence modeling aids developers in crafting functional software by offering insights into the semantic meaning of sentences. Traditional approaches to information retrieval assess data based on individual words or characteristics, which often overlooks the essence of the analyzed statement. In 2014, Kalchbrenner illustrated a dynamic CNN approach with k-max pooling in the training context (Kalchbrenner et al. 2014), as depicted in Fig. 6. A word embedding size of 4 is utilized, and the network comprises two convolutional layers, each with distinct feature maps. The filter widths for the upper and lower layers are 3 and 2, respectively.

Through this methodology, connections among words can be discerned sans reliance on dictionaries or parsers (Gidaris and Komodakis 2015). Furthermore, Collobert and Weston have devised a variant of convolutional neural network (CNN) capable of simultaneous execution of diverse natural language processing tasks, encompassing language modeling, chunking, named entity recognition, and semantic role labeling (Collobert 2011). Integration of a Max-pooling mechanism enables the extraction of intricate details from various segments of the document. Additionally, to refine precision and reduce model dimensions, an intermediate layer (bottleneck) is introduced to facilitate the acquisition of succinct document representations. Moreover, instead of feeding low-dimensional word vectors into CNNs, the approach described in Liu et al. (2017) involves training high-dimensional text embeddings for smaller text categories.

The efficacy of both word embeddings and CNN architectures has been investigated concerning their impact on model performance. The VDCNN model, as proposed by Conneau et al. (2016), operates by directly processing individual characters through small convolutions and pooling operations. Results indicate that VDCCN's performance improves

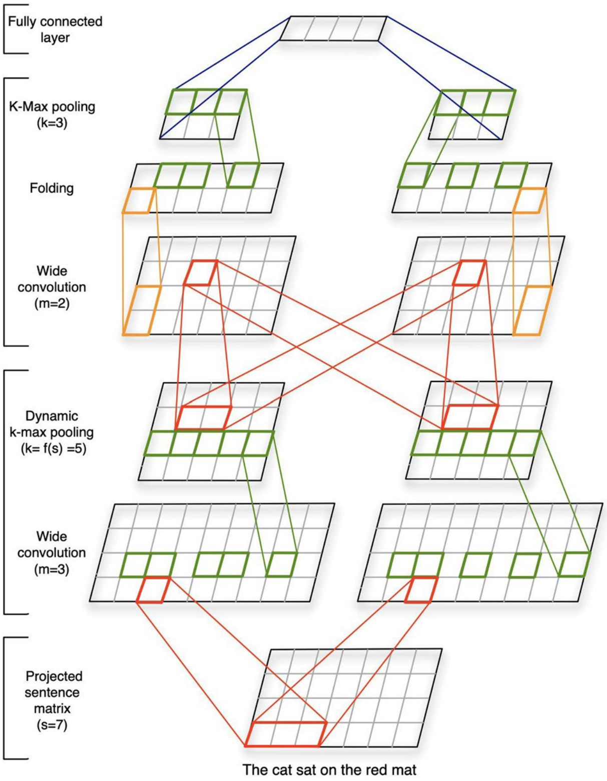

Fully connected layer
K-Max pooling (k=3)
Folding
Wide convolution (m=2)
Dynamic k-max pooling (k= f(s) =5)
Wide convolution (m=3)
Projected sentence matrix (s=7)
The cat sat on the red mat

Fig. 6 A typical DCNN design for the 7 word input sentence (Kalchbrenner et al. 2014)

with increased network depth. Authors in Duque et al. (2019) adapted VDCNN's structure to accommodate the limitations of mobile platforms.

6.4 Attention-based models

The human attention mechanism can be categorized into two distinct classes (Tsotsos et al. 1995). Saliency-based attention, characterized by its external-to-internal processing, constitutes the initial form of unconscious attention. For example, individuals are more likely to perceive voices in a crowded environment if they are emitted loudly. In the context of deep learning, this resembles the max-pooling and gating mechanism (Hochreiter and Schmidhuber 1997; Cho et al. 2014), wherein less suitable values (e.g., smaller ones) are suppressed while more relevant ones are retained. The other attentional class, termed "focused attention," operates in a top-down manner. This attentional mode is directed towards a specific objective or set of activities, facilitating deliberate and purposeful concentration. The majority of attention mechanisms in deep learning are tailored for singular tasks.

In neural machine translation, neural networks are employed for the task of translating text from one language to another. A significant obstacle in this process is aligning sentences across different languages, particularly with longer sentences. To address this challenge and improve translation quality, researchers in Bahdanau et al. (2014) introduced the attention mechanism into neural networks. This mechanism enables the network to selectively focus on specific sections of the source text during translation. Since then, various enhancements have been proposed, such as local attention (Luong et al. 2015), supervised attention (Mi et al. 2016; Liu et al. 2016), hierarchical attention (Zhao et al. 2018), and self-attention (Vaswani et al. 2017; Yang et al. 2018). These enhancements aim to better align words and enhance translation performance by experimenting with different attention architectures. Figure 7 illustrates the method proposed by the authors in Yang et al. (2018), incorporating a window size of 2.

Labeling texts is the primary objective of text classification, a process widely utilized in various applications including topic categorization (Wang et al. 2012), sentiment analysis (Maas et al. 2011; Pang et al. 2008), and spam identification (Sahami et al. 1998). Self-attention mechanisms are predominantly employed to enhance the representation of documents in these classification endeavors (Letarte et al. 2018; Shen et al. 2018; Lin et al. 2017). Consequently, numerous studies have integrated self-attention with other attention techniques, such as hierarchical self-attention (Yang et al. 2016) and multi-dimensional self-attention (Lin et al. 2017). Architectures incorporating attention models, including transformer (Song et al. 2019; Ambartsoumian and Popowich 2018) and memory networks (Tang et al. 2016; Zhu and Qian 2018), have also been applied in these tasks.

In addition to question answering, document retrieval, entailment classification, paraphrase detection, and recommendation systems based on reviews, text matching represents a significant area of examination in natural language processing and information retrieval. Various innovative techniques, such as memory networks (Sukhbaatar et al. 2015), attention over attention (Cui et al. 2016), inner attention (Wang et al. 2016), structured attention (Kim et al. 2017), and co-attention (Tay et al. 2018; Lu et al. 2016), have been developed in conjunction with attention mechanisms to address this research domain.

6.5 Knowledge graph (KG) based models

The integration of KG data into neural QA systems is currently a subject of intense study. Some research investigates the utilization of two-tower models (Wang et al. 2019) that merge a graph-based knowledge representation and a language-based representation without any form of interaction between these two components. There are also studies that attempt to utilize one modality to ground the other, such as Knowledgeable Reader (Mihaylov and Frank 2018), KagNet (Lin et al. 2019), and KT-NET (Yang et al. 2019), employing an encoded version for a linked KG in order to enhance the textual expression for any of the question-answering instances. Certain methods, like MHGRN (Feng et al. 2020) and Lv et al.'s graph reasoning model (Lv et al. 2020), demonstrate a different information flow in their approach. They achieve this by employing a textual representation while working

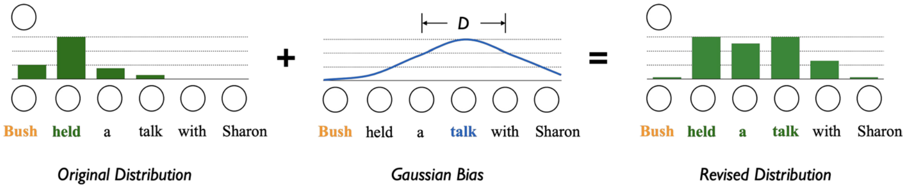

D
+
=
Bush held a talk with Sharon
Bush held a talk with Sharon
Bush held a talk with Sharon
Original Distribution
Gaussian Bias
Revised Distribution

Fig. 7 Illustration of the proposed method of modeling locality in Yang et al. (2018)

with an extracted KG for the given example. However, in all of these contexts, interaction between the two modalities is constrained since information can only go in one direction.

Recent approaches explore broader integrations of the two modalities. To construct local KGs suitable for Question Answering (QA) (Wang et al. 2020; Hwang et al. 2021), certain methodologies aim to extract implicit information ingrained in LMs by leveraging training on structured KG data (Bosselut et al. 2019; Petroni et al. 2019; Hwang et al. 2021). However, following LM training on factual data, several techniques discard the static KG, thereby forfeiting valuable structural cues essential for guiding reasoning. A recent advancement, QA-GNN (Ren et al. 2021), advocates employing message propagation to concurrently refine both LM and Graph Neural Network (GNN) representations. This approach is facilitated by the initiation of the textual component within this amalgamated structure.

Additionally, other studies explore the viability of pre-training knowledge graphs in conjunction with language models. However, akin to question answering (QA), modality interaction is frequently confined to feeding knowledge into language (Zhang et al. 2019; Shen et al. 2020; Donghan et al. 2022), rather than fostering interactions across multiple layers. The authors of Zhang et al. (2019) have presented a work that closely resembles this, albeit they limit the system's flexibility by utilizing identical modality configurations for both the LM and KG.

7 Applications of LLMs

Recent advancements in deep learning, in conjunction with numerous PLMs, facilitate the efficient execution of various NLP tasks. To leverage LLMs, tasks can be reformulated as text generation challenges, enabling the application of LLMs to efficiently address these tasks.

7.1 Text generation

Text generation is a critical application of language models (LMs), aiming to generate word sequences based on input data. The diversity of objectives and initial materials introduces numerous challenges in text production. For instance, in automated speech recognition (ASR), a sequence of spoken words serves as input, while a sequence of written words is the corresponding output. Similarly, machine translation involves utilizing an input text sequence along with the target language to generate a text sequence in the target language. Generating a story exemplifies the task of generating text from a given topic. Decoding plays a pivotal role in text generation by determining the next linguistic unit in the output sequence. Effective decoding techniques should generate coherent continuations given a context. The significance of decoding techniques has grown in parallel with the increasing complexity of LMs.

The maximization-oriented decoding technique endeavors to identify tokens with the highest probability of text generation, operating under the assumption of model accuracy. In greedy search methodologies (Zhao et al. 2017; Xu et al. 2017), the subsequent token selection consistently prioritizes the one with the greatest probability. Conversely, in beam search approaches (Li et al. 2016; Vijayakumar et al. 2018; Kulikov et al. 2018), tokens with the highest probabilities are retained iteratively, favoring sequences with the highest cumulative probability, thereby avoiding overlooking plausible tokens with lower probabilities. Recent advancements in decoding algorithms have introduced trainable methodologies, such as trainable greedy decoding in neural machine translation (Gu et al. 2017), wherein reinforcement learning optimizes decoding objectives to attain the best translation, albeit potentially resulting in low-probability token generation due to sampling from an unstable tail distribution. The consequence of unrelated prefixes generating nonsensical outcomes underscores recent proposals like Top-k sampling (Fan et al. 2018) and Nucleus sampling (Holtzman et al. 2019), both employing truncated language model distributions to bias sampling towards the most probable tokens. Diverse Beam Search (DBS) (Vijayakumar et al. 2018) builds upon Beam search as a sampling-based decoding algorithm, potentially trainable, with beam diversity settings tailored for different inputs or tasks using reinforcement learning.

7.2 Vision-language models

Researchers have aimed to develop comprehensive models that integrate two distinct types of data, referred to as Vision-Language Models (VLMs).The conceptual framework of these models draws inspiration from the efficacy demonstrated by pre-trained models within the realms of CV and NLP (Wen et al. 2023). VLMs can be classified using either fusion-encoder models or dual-encoder models. Fusion-encoder models employ multi-layer cross-modal Transformer encoders to jointly encode image and text pairs, combining their visual and textual representations. Conversely, dual-encoder models independently encode images and text. The interactions between the two modalities are then captured using a dot product or a multi-layer perceptron.

The encoder modules (fusion) of these models receive both visual features and text embeddings as input and employ various fusion techniques to accurately capture the interaction between the visual and textual modalities. Following self-attention or cross-attention operations, the latent features of the top layer are considered a merged representation of the multiple modalities. VisualBERT (Li et al. 2019) represents a significant advancement, using self-attention to implicitly align textual components with regions in the corresponding image, as depicted in Fig. 8. By combining image regions and a Transformer, the self-attention mechanism can learn latent correspondences between words and images. Initially, it is pre-trained on caption data using masked language modeling and sentence-image prediction tasks, and subsequently fine-tuned for specific tasks. It integrates BERT (Devlin et al. 2018) for processing linguistic data and pretrained Faster RCNN (Ren et al.

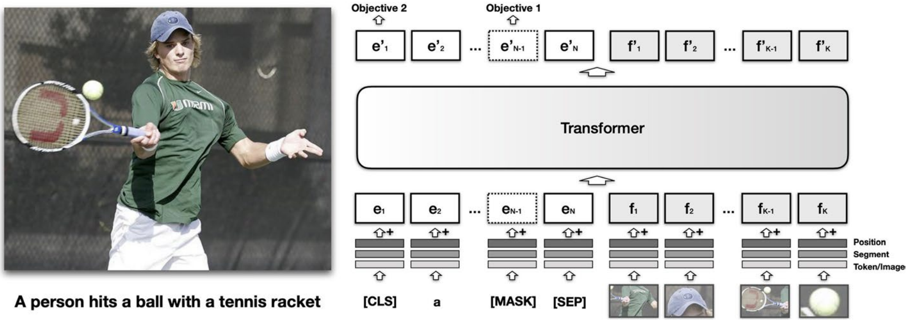


Fig. 8 Methodology for VisualBERT (Li et al. 2019)

2015) for generating object proposals. To capture complex relationships, the original text and image data from item proposals are fed into VisualBERT as unordered input tokens and processed simultaneously by various Transformer layers. Following this development, a variety of VLM models, including Uniter (Chen et al. 2020), OSCAR (Li et al. 2020), and InterBert (Lin et al. 2020), utilized BERT as a textual encoder and Faster-RCNN as an object proposal generator to model the dynamic relationship between visual information and language.

Dual-stream architectures integrate a cross-attention mechanism to capture interactions between visual and verbal elements, contrasting with the single-stream designs that manage a single information stream. Typically, the cross-attention layer comprises two unidirectional sub-layers: one processes information from vision to language, and the other processes information from language to vision. These sub-layers enable communication and coordination between the two modalities. For example, ViLBERT (Lu et al. 2019) utilized co-attentional transformer layers to enable cross-modal information sharing by independently processing visual and textual inputs. Subsequent research, such as LXMERT (Tan and Bansal 2019), Visual Parsing (Xue et al. 2021), ALBEF (Li et al. 2021), and WenLan (Huo et al. 2021), further refined this approach by employing distinct transformers before cross-attention to separate intra-modal and cross-modal interactions. Chen et al. (2022) proposed VisualGPT to manage limited image-text data within the same domain by introducing a novel self-resurrecting encoder-decoder attention mechanism into PLMs.

7.3 Personalised learning

Educators can utilize AI to analyze student performance and behavior data, identify areas of difficulty, and provide personalized guidance for addressing these issues. AI tools are employed to develop adaptive learning systems that adjust the difficulty of assignments and assessments based on each student's needs and abilities, thereby facilitating a more personalized learning experience and a more accurate assessment of student progress. This approach ensures that students are appropriately challenged without being overwhelmed, which has been demonstrated to enhance both motivation and engagement (Baars et al. 2022). AI provides personalized feedback that highlights specific areas for improvement and suggests strategies, enabling students to better understand their strengths and weaknesses and develop effective study habits. This individualized learning experience and targeted instruction significantly enhance student support and foster a more productive learning environment. By providing timely feedback and assistance, AI and NLP contribute to the development of students' metacognitive skills, including the ability to reflect on their learning processes and formulate improvement plans (Khan et al. 2023).

Furthermore, AI contributes to the creation of personalized learning plans by considering each student's learning style, interests, and goals, thereby maintaining student motivation and engagement, which ultimately leads to enhanced academic outcomes (Paranjape et al. 2019; Mbakwe et al. 2023). Additionally, as educational research advances, AI can keep pace with the latest effective teaching and learning strategies and methodologies. Consequently, students can adopt an educational approach that is both innovative and effective. In the long term, AI has the potential to transform the provision of individualized instruction and support by teachers, thereby improving student achievement. The scarcity of experienced tutors and the cost associated with traditional tutoring services, which often require hourly fees, can be prohibitive for students. For instance, GPT-4 addresses these issues by offering prompt access to accurate answers and comprehensive explanations

(Mbakwe et al. 2023). Chatbots powered by NLP can provide immediate responses to straightforward questions, thereby democratizing access to foundational information (Ashwini et al. 2022; Libbrecht et al. 2020).

It is essential to balance the use of AI technology with human involvement in the educational process. While AI can serve as a valuable tool for educators and learners, the significance of human interaction should not be underestimated (Vasileva and Balyasnikova 2019). The influence of pro-social emotions and empathy on student performance is considerable. Consequently, it is imperative to assess and integrate an optimal level of human participation in instructional strategies that incorporate AI. Educators can enhance student retention and achievement by strategically utilizing technologies such as chatbots to foster a sense of community among students. AI is expected to have extensive impacts on the field of pharmacy and education in general. In the near future, classrooms tailored to individual student needs, interests, and preferences may become standard. Educators must remain receptive to new ideas and recognize technology as an essential tool to capitalize on the current educational paradigm shift.

7.4 Code generation and code completion

Generating new code and comprehending existing code are the two primary categories of software naturalness-related AI-assisted programming activities. Instances of the former encompass code generation, code auto-completion, code translation process, code refinement, along with code summarization. The latter involves identifying errors and detecting code duplicates, among other tasks. Researchers have focused extensively on the methodologies of current language models (LMs) utilized for specific tasks, such as code generation. They have achieved progress by improving pre-training techniques, expanding the training datasets, fine-tuning datasets, and developing more effective evaluation criteria for these tasks.

Using user-specified constraints, program synthesis (or source code generation) automatically generates source code in a programming language (Waldinger et al. 1969; Manna and Waldinger 1971). The earliest documented instance of code generation involved utilizing software for theorem proving to construct a proof based on user-provided specifications, followed by the extraction of logical programs corresponding to that proof (Manna and Waldinger 1975; Green 1981). DL methods, such as Long Short-Term Memory (LSTM) (Dong and Lapata 2016) and Recursive Reverse-Recursive Neural Network (Parisotto et al. 2016), have utilized neural approaches to construct output programs with predefined inductive biases using a large number of program samples, as illustrated in Fig. 9. In recent years, transformer-based LLMs such as GPT-3 (Brown et al. 2020) and T5 (Raffel et al. 2020) have demonstrated remarkable success in the code generation tasks. These models utilize the contextual representations derived from extensive code samples, along with publicly available code repositories and natural language data, significantly enhancing the program synthesis process. The incorporation of systematic pre-training and fine-tuning approaches enables a thorough comprehension of code organization and intrinsic semantics, rendering these methods well-suited for software development endeavors.

Code completion assists programmers by providing text input suggestions as they write code, serving as a development tool particularly beneficial for novice programmers. This process, also known as autocompletion (Dong et al. 2022), aims to expedite development and reduce errors by proposing suitable names for variables, methods, and even the entire code segments. Initial examinations into code completion utilized statistical language

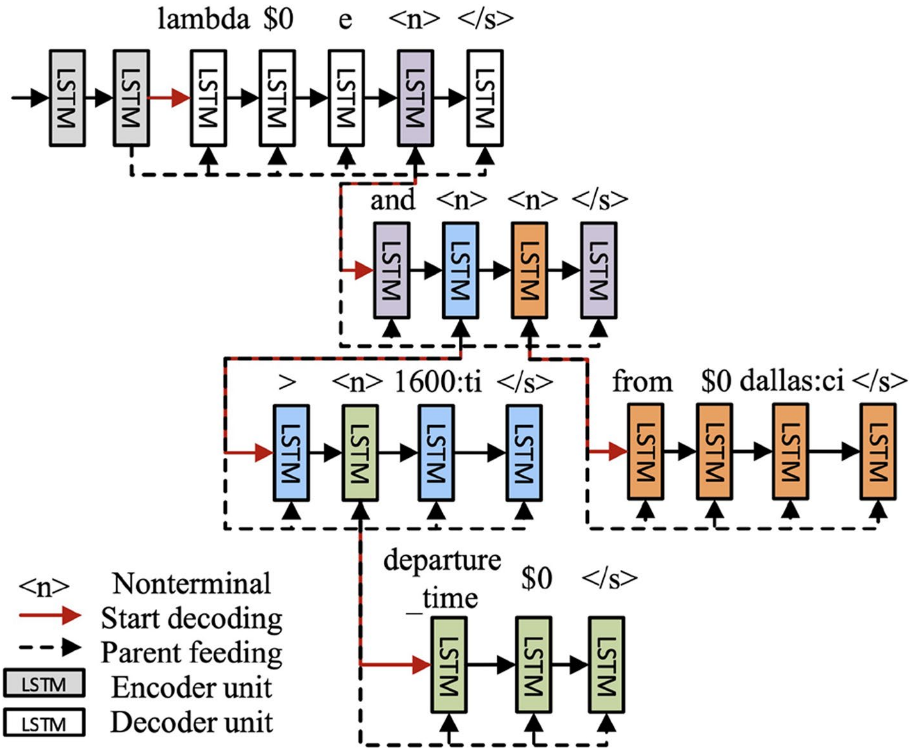


Fig. 9 Hierarchical tree decoder (Dong and Lapata 2016) in a sequence-to-tree (SEQ2TREE) model

models (Robbes and Lanza 2008; Bruch et al. 2009) to examine the semantic content within source code, irrespective of its syntactic arrangement. To achieve this, LSTM-based deep learning algorithms (Bruch et al. 2009) were employed. However, the limitations of LSTM-based models prompted the development of the transformer architecture, which has since become integral to code completion systems. In the course of training, language models for code completion commonly utilize a causal language model to predict the subsequent token in a sequence of recognized tokens. Recent progressions in employing LLMs for code completion have exhibited enhanced efficacy on evaluation benchmarks such as CodeXGLUE (Lu et al. 2021), surpassing traditional statistical language models and earlier DL approaches.

7.5 Biomedicine

The application of language models is broadening in the biological sciences, encompassing both fundamental biomedical research and clinical healthcare support. LLMs have the capacity to undergo training for the purpose of scrutinizing and predicting biological functionalities, mechanisms of diseases, and procedures related to drug development through the utilization of genetic and proteomic datasets. Prediction of protein structure and interactions is crucial for understanding biological processes and developing new medications, and LLMs can assist in this area. In the domain of clinical healthcare assistance, the utilization of NLP techniques on medical records facilitates the detection of patterns, diagnosis

Table 7 Summary of recent studies investigating the implementation of Language Models (LLMs) in various fields

Name	Year	Methodology	Dataset	Architecture	Metric	Domain	Findings
UniLM (Dong et al. 2019)	2019	Self attention	GLUE	Shared transformer network	BLEU-4, METEOR, ROUGE-L	NLU, NLG	Used a centralized Transformer network with self-attention masks to determine which variables to predict.
APPS (Hendrycks et al. 2021)	2021	Python	Open access coding platforms	Transformer decoder	Accuracy, BLEU	NLU, Code generation	The dataset evaluates models based on their capacity to convert a free-form natural language specification into usable Python code.
PET (Schick et al. 2021)	2021	Semi-supervised training	Yelp Reviews, AG's News, Yahoo Questions	Transformer encoder	Accuracy, F1-Score	Text Classification, Langauge Inference	To better understand a task, the model reformats input examples as cloze-style phrases
MST (Li et al. 2021)	2021	Attention-guided masking	ImageNet 1k	CNN, Transformer	Accuracy, AP, mIoU	Token prediction	This model can preserve the global semantic information of an image while explicitly capturing the local context.
MoCL (Sun et al. 2022)	2021	Unsupervised learning	Molecular datasets	GNN	Confusion matrix, AUC	Hybrid	Maximizing mutual information between paired graph augmentations has proven effective on downstream tasks

Table 7 (continued)

Name	Year	Methodology	Dataset	Architecture	Metric	Domain	Findings
Chinchilla (Hoffmann et al. 2022)	2022	Scaling model size	BookCorpus, Enwik8, WebText	Transformer	Parametric fit	Text generation, Translation, Question answering.	Adjustment in the size of model along with no. of training tokens accomplishes compute-optimal training
MAE (He and Chen 2021)	2022	Random masking	COCO, ADE20K	Transformer	Accuracy, mIoU	Pixel reconstruction (CV)	In order to reconstruct the missing pixels, the model masks out random areas of the input image
CAE (Chen et al. 2022)	2022	Random partitioning	Food-101, Clipart, Sketch	Transformer	Accuracy, mIoU	Pixel reconstruction (CV)	Benefits of representation learning can be seen empirically because predictions tend to be located in the space of encoded representations.
MGAE (Tan et al. 2022)	2022	Self-supervised learning	ogbl-ddi, ogbl-collab, and ogbl-ppa	GCN	AP, AUC	Masked edge recnostruction (GL)	During training, model randomly masks most edges and try for reconstructing graph edges
HGMAE (Tian et al. 2023)	2022	Generative self-supervised learning	DBLP, Freebase, AMiner	Transformer	AUC, Ma-F1, Mi-F1	Metapath Masking Reconstruction	Efficient and stable heterogeneous graph learning is achieved through the use of metapath masking, adaptive attribute masking, and a dynamic mask rate.

Table 7 (continued)

Name	Year	Methodology	Dataset	Architecture	Metric	Domain	Findings
BEiT (Bao et al. 2022)	2022	Self-supervised learning	ImageNet-1K, ADE20K	Transformer	mIoU	Token prediction	Model tokenizes image first. It randomly masked image patches and fed them to the backbone Transformer
OPT (Zhang et al. 2022)	2022	Pre-trained transformers	BookCorpus, CommonCrawl, Wikipedia	Transformer decoder	Accuracy	Causal language modeling	Pre-trained transformers for use only with the decoder, spanning parameters from 125 M to 175B
PeCo (Dong et al. 2022)	2023	dVAE training	ImageNet-1K, ADE20K	Transformer	mIoU	Pixel reconstruction (CV)	Learned visual tokens have better semantic meanings, helping pre-training perform better in downstream tasks.
SAM (Kirillov et al. 2023)	2023	Zero-shot generalization	SA-1B	Transformer	mIoU, AP	Reconstruction, Segmentation (CV)	The promptable model can apply zero-shot to new image distributions and tasks

NLU natural language understanding, *NLG* natural language generation, *CV* computer vision, *GCN* graph convolutional network, *GL* graph learning, *GNN* graph neural network) of conditions, and provision of tailored treatment recommendations. This is accomplished through the employment of LLMs that have undergone pre-training or fine-tuning with medical datasets.

Recent advancements in LLMs have displayed potential in fundamental biomedical examination (Mahjour et al. 2023). These models have the capability to integrate data from multiple domains such as molecular structures, genetics, proteomics, and metabolic pathways, thereby providing a broader comprehension of biological systems. Within the domain of molecular and biological sciences, bespoke LLMs tailored to specific applications have emerged. For instance, to integrate spatial information into molecular data, researchers can utilize tools like MoLFormer (Ross et al. 2022), a high-capacity molecular SMILES transformer model incorporating relative position embedding. Another example is Nucleotide Transformer (Dalla-Torre et al. 2023), a pretrained foundational model leveraging LLMs to learn from biological sequences, thus facilitating more precise predictions of molecular phenotypes. Additionally, the evolutionary scale modeling (ESM) family (Rives et al. 2021), comprising transformer protein language models such as ESM-2 and ESM-Fold, surpasses earlier single-sequence protein language models, demonstrating enhanced capability in generating accurate structure predictions based on protein sequences. ProtGPT2 (Ferruz et al. 2022) aims to generate de novo protein sequences adhering to natural principles, having been specifically trained in the domain of proteins. Furthermore, the ProGen (Madani et al. 2023) language model employs deep learning techniques to produce protein sequences with predefined functionalities. Notably, synthetic proteins optimized for specific families of lysozymes using ProGen have exhibited catalytic efficiencies comparable to their naturally occurring counterparts with low sequence identity.

LLMs have significantly contributed to the understanding of medical texts and electronic health data, offering numerous potential benefits for healthcare (Rao et al. 2023, 2023; Zuccon and Koopman 2023). Recently, several large-scale LLMs specialized for specific NLP tasks in the biological domain have emerged. One such model, BioGPT (Luo et al. 2022), a generative Transformer language model, demonstrates exceptional performance across various biomedical NLP tasks, including document classification task, relation extraction task, and question answering task, owing to its training on an extensive biomedical literature corpus.

8 Challenges and future research directions

Recent advancements have significantly progressed the creation of language models applicable to NLP. Understanding the capabilities and limitations of these LLMs is crucial for maximizing their utility in NLP tasks. Although LLMs have proven effective in various NLP applications, obstacles remain due to the intricacies of language and computational demands. This section delineates significant obstacles and suggests prospective avenues for additional examination within this field.

8.1 Hallucination

One significant constraint of current systems involves their susceptibility to exploitation through the generation of plausible false assertions by LLMs. Recent examinations suggest that through optimizing the utilization of existing model capabilities, this issue may potentially be resolved. The precision with which LLMs discern true statements improves proportionally with model scale (Burns et al. 2022; Kadavath et al. 2022). Recent methodologies demonstrate a significant reduction in explicit bias and toxicity within model outputs (Dinan et al. 2019; Bai et al. 2022; Ganguli et al. 2023), primarily by leveraging the models' ability to identify such undesirable behaviors upon inquiry. Although the efficacy of these countermeasures in entirety remains uncertain, they are expected to diminish the occurrence and prominence of problematic behaviors over time. Noteworthy failure modes might evade partial solutions. For example, due to concerns regarding sandbagging, simplistic efforts to mitigate hallucination are prone to silent failures, which may falsely enhance their credibility. Employing traditional techniques to instruct a future LLM to adhere to truthfulness, and if said LLM can reasonably anticipate which factual claims are likely to be scrutinized by human annotators, then training it to uphold truth solely in verifiable claims becomes feasible.

8.2 Computational requirements

The process of training a LLM, comprising millions (sometimes billions) of parameters, typically imposes substantial computational demands. Managing vast datasets within code constitutes one facet of this process, while optimizing the model's parameters represents another crucial aspect aimed at ensuring prediction accuracy. Consequently, the training process becomes highly resource-intensive (Zhang and He 2020). Insufficient training data and limited computational resources, such as GPUs, electricity, or memory, can result in significant escalations in computational costs. Nevertheless, the quality of the training data utilized in language model training holds paramount importance, as inadequate data quality or bias can lead to imprecise predictions. The utilization of LLMs for training, fine-tuning, and execution necessitates considerable computational resources, posing challenges for companies operating with constrained hardware capabilities (Han et al. 2021). Researchers and developers have access to various techniques to alleviate the computational burden associated with training LLMs. These techniques encompass working with data subsets (Liang and Zou 2022), fine-tuning hyperparameters (Yin et al. 2021), and employing transfer learning to leverage insights gained from previous tasks. These methods have the potential to expedite the training process and alleviate strain on the requisite computer systems.

8.3 Ethical alignment and privacy concerns

It is imperative to ensure that increasingly intricate and self-reliant models adhere to human values and objectives. Methods need to be devised to guarantee these models function as intended and avoid favoring specific outcomes inappropriately. Alignment techniques should be integrated at an early stage in model development. The capacity to perceive and comprehend the inner workings of the model is also pivotal for evaluating and maintaining coherence. The alignment of posthuman systems poses an even more formidable challenge for the future. The convergence of these advanced systems may introduce novel complexities and ethical considerations (Bai et al. 2022; Bowman et al. 2022), thus, while they may currently surpass our requisites, it is nonetheless crucial to contemplate and prepare for them.

Few of the studies examined or addressed privacy concerns regarding innovations based on LLMs. None of the studies optimizing LLMs with textual data provided by students have disclosed their strategies for obtaining consent (e.g., whether students are informed about the collection and intended use of their data) and measures for data protection (e.g., anonymization and sanitization of data). Given that LLM-based innovations operate with stakeholders' natural languages, which may contain personal and sensitive information about their private lives and identities (Brown et al. 2022), the lack of attention to privacy issues is particularly troubling. Since the consent process is often integrated into the enrollment or sign-up procedures of these platforms (Tsai and Gasevic 2017), stakeholders may not realize that their textual data (e.g., forum posts or conversations) on digital platforms (e.g., MOOCs and Learning Management Systems) is being utilized in LLM-based innovations for various automation purposes (e.g., automated replies and training chatbots). An informed consent process would not resemble this approach.

Technologies utilizing "narrow" Artificial Intelligence (AI) are presently ubiquitous across various aspects of daily routines. The anticipated advancement in artificial intelligence is recognized as artificial general intelligence (AGI). Unlike narrow AI, which is confined to singular tasks, AGI is anticipated to possess the capacity for learning, adaptation, and modification of its functional repertoire, enabling engagement in tasks not originally programmed (Nick 2014; Everitt et al. 2018; Gurkaynak et al. 2016). Speculation regarding the hazards associated with AGI exists despite its potential for extensive and profound benefits (Amodei et al. 2016; Nick 2014; Brundage et al. 2018; Salmon et al. 2021). These hazards may emanate from dysfunctional AGI, malevolent design or utilization, or even from a weak or "superintelligent" AGI striving for goal achievement, potentially at the expense of other domains (McLean et al. 2023; Critch and Krueger 2020; Salmon et al. 2021). If AGI presents an existential peril to humanity, one strategy to alleviate such risk could involve ensuring alignment between AGI values and human values. A prevalent depiction of AI alignment entails a collaborative progression between humans and AGIs aimed at shared objectives. However, presuming consistent articulation of "values" across individuals, which underpins the notion of "alignment," entails a risk. In decision theory, values manifest as utilities, and alignment occurs when two parties act to enhance their respective utilities. Consequently, the divergence in utilities signifies one facet of existential risk (Salmon et al. 2023). For instance, contemplate Bostrom's paper-clip maximizer, where the objective is to optimize paper-clip production regardless of Earth's resource depletion (Nick 2014).

8.4 Limited influence over LLM models

Forecasts regarding the potential functionalities of forthcoming Large Language Models (LLMs), derived from the economic incentives, values, or inclinations of their developers, are prone to inadequacy. This is primarily due to the emergent nature and inherent unpredictability of many significant LLM capabilities, alongside the limited influence wielded by LLM developers over the specific capabilities that future iterations will possess. As exemplified by GPT-4, while it manifested several desired abilities as envisioned by its designers, it also exhibited certain undesirable traits initially, such as providing instructions on the synthesis of biological weapons to non-specialists, prompting substantial efforts from its creators to rectify such behaviors [13]. Moreover, developers typically possess only a superficial grasp of an LLM's functionalities when deciding on its deployment. There remains a possibility for users to elicit fundamentally novel behaviors from LLMs, unforeseen by developers, through causal reasoning processes akin to those observed in GPT-3, defying current evaluation or analytical methodologies.

8.5 Data deficiency and modalities

It is noteworthy that the majority of pre-trained datasets are limited to single modes or languages. The advancement of LLMs significantly depends on the existence of pre-trained datasets that accommodate diverse data types such as multimodal, multilingual, and graph data. Presently, technical obstacles arise due to the distinctive characteristics of such data. Unlike NLP and CV, most nodes and edges in graph data lack substantial amounts of unlabeled data suitable for pretraining. Nevertheless, there are a few reusable nodes in molecular and protein networks that deviate from this data scarcity pattern. However, examination of the initial training phase of graph models remains at an early stage. Data obtained from the Internet of Things (IoT) is expected to be abundant and filled with real-world data. For instance, sensor data from inertial measurement units can capture details about users' social activities (Wang et al. 2018; Han et al. 2019). There is an urgent need for further research to strengthen the theoretical foundations and delve into diverse interpretations of various pretext tasks.

Recent research has explored PFMs incorporating diverse media modalities, including text paired with images, text combined with audio, and similar configurations, predominantly focusing on bimodal setups. Facilitating the training of multimodal PFMs necessitates establishing intermodal connections, highlighting the exigency for novel multimodal datasets. Consequently, the creation of such datasets emerges as a pressing concern. The extant multilingual PFM addresses the challenge of language-specific resource scarcity, facilitating advancements in state-of-the-art technologies such as text summarization, various question-answering software, neural machine translation (NMT) systems, among others. However, the current PFM remains constrained to a mask LM. Introducing pertinent new tasks can enhance the efficacy of multi-LMs. Furthermore, the transition from single-lingual to multi-lingual vocabularies entails a substantial increase in the number of model parameters requiring learning.

9 Discussion

This research entails a comprehensive examination of LLMs, encompassing an in-depth examination into various facets within the domain. Commencing with elucidating cutting-edge concepts pertaining to LLMs, this study elucidates the contextual framework underpinning the operation of LLM-based models. Subsequent sections meticulously scrutinize pivotal components, particularly CLMs and PLMs, with specific attention to their structural architecture, learning methodologies, and the linguistic units they process. A detailed scrutiny is undertaken to assess the pivotal role of training datasets, emphasizing their substantial influence on the efficacy of LLMs. Delving into the domain of word embeddings, this analysis assesses methodologies such as Latent Semantic Analysis (LSA), Word2Vec, Global vectors (Glove), FastText, and Contextualized Word Embeddings (CoVe) to elucidate their distinct contributions to the comprehension and representation of language. A comprehensive examination of recent advancements in deep learning methods pertinent to LLM tasks is provided. This segment scrutinizes architectural frameworks critical for a diverse array of applications, encompassing text generation, vision-language models, personalized learning, and code generation. These frameworks encompass Feed Forward Neural Networks, RNN, CNN, Attention-based models, and KG based models. Applications extend to the domain of biomedicine as well. Nonetheless, this study not only accentuates the accomplishments of LLMs but also conscientiously addresses inherent challenges while proposing potential avenues for further examination. Topics such as hallucination, computational requisites, ethical considerations, privacy apprehensions, limited control over LLM models, and challenges associated with inadequate data in diverse formats are meticulously examined. This comprehensive analysis of challenges serves as a prelude to future explorations and advancements in the expansive domain of language models. In summation, this research offers a scholarly examination of the intricate nature of language models, encompassing foundational components, varied applications, and anticipated obstacles.

Nevertheless, it is crucial to recognize the inherent constraints within this examination. The extent of this research, though thorough, is not all-encompassing, and there could be emerging frameworks or theories not accounted for. Moreover, the fluidity of the field suggests that specific findings might undergo modifications with further progress. Additionally, the research primarily concentrates on established methodologies and might not encompass recent developments in the swiftly evolving domain of language models.

10 Conclusion

Recent advancements in LLMs have significantly impacted NLP. Understanding the capabilities and limitations of LLMs across various NLP tasks is crucial for their efficient utilization. This paper conducts a thorough analysis and comparative assessment of existing literature on LLM-based methodologies across diverse domains including video, image, audio, and text processing. Additionally, it provides a detailed examination of LLM components such as linguistic units, training datasets, word embeddings, and the methodologies employed in both CLMs and PLMs. The study commences by investigating factors influencing the effectiveness of prominent models, encompassing architectures akin to GPT-style and BERT-style designs. Subsequently, it explores the application of LLMs in tasks spanning vision and language integration, personalized learning, text generation, biomedical applications, and code generation. This paper caters to individuals seeking an introduction to LLMs as well as those already familiar with the subject matter. It offers a scholarly analysis of the intricate nature of language models, acknowledging inherent limitations. Covering foundational elements, diverse applications, anticipated challenges, and acknowledged constraints, this work serves as a valuable resource for scholars, practitioners, and stakeholders navigating the evolving landscape of language processing technologies. The work aims to assist researchers and practitioners in unlocking the full potential of LLMs and driving advancements in language technology.

Acknowledgements N/A.

Author Contributions Pranjal Kumar conceptualized and wrote the manuscript. Reviewed and approved the final version of the manuscript.

Funding N/A.

Data availability statement N/A.

Declarations

Conflict of interest The authors declare no Conflict of interest.

Ethical approval N/A.

Consent for publishing Yes.

References

Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman FL, McGrew B (2023) Gpt-4 technical report. arXiv preprint arXiv:2303.08774

Al-Hashedi A, Al-Fuhaidi B, Mohsen AM, Ali Y, Gamal Al-Kaf HA, Al-Sorori W, Maqtary N (2022) Ensemble classifiers for Arabic sentiment analysis of social network (twitter data) towards covid-19-related conspiracy theories. Appl Comput Intell Soft Comput 2022:1–10

Al-Rfou R, Choe D, Constant N, Guo M, Jones L (2019) Character-level language modeling with deeper self-attention. Proc AAAI Confer Artif Intell 33:3159–3166

Ambartsoumian A, Popowich F (2018) Self-attention: a better building block for sentiment analysis neural network classifiers. arXiv preprint arXiv:1812.07860

Amodei D, Olah C, Steinhardt J, Christiano P, Schulman J, Mané D (2016) Concrete problems in ai safety. arXiv preprint arXiv:1606.06565

Andrabi SA, Wahid A (2022) A comparative study of word embedding techniques in natural language processing. In: Computational vision and bio-inspired computing: proceedings of ICCVBIC 2021. Springer, pp 701–712

Ashwini S, Rajalakshmi NR, Jayakumar L et al (2022) Dynamic NLP enabled chatbot for rural health care in India. In: 2022 2nd international conference on computer science, engineering and applications (ICCSEA). IEEE, pp 1–6

Asudani, DS, Nagwani NK, Singh P (2023) Impact of word embedding models on text analytics in deep learning environment: a review. Artif Intell Rev 56(9):10345–10425

Baars M, Khare S, Ridderstap L (2022) Exploring students' use of a mobile application to support their self-regulated learning processes. Front Psychol 13:793002

Badri N, Kboubi F, Chaibi AH (2022) Combining fasttext and glove word embedding for offensive and hate speech text detection. Proc Comput Sci 207:769–778

Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473

Bai Y, Kadavath S, Kundu S, Askell A, Kernion J, Jones A, Chen A, Goldie A, Mirhoseini A, McKinnon C, Chen C (2022) Constitutional ai: Harmlessness from ai feedback. arXiv preprint arXiv:2212.08073

Bao H, Dong L, Piao S, Wei, F BEiT: BERT Pre-Training of Image Transformers. In: International Conference on Learning Representations

Bengio Y, Ducharme R, Vincent P (2000) A neural probabilistic language model. Adv Neural Inf Process Syst 13

Bhatt D, Patel C, Talsania H, Patel J, Vaghela R, Pandya S, Modi K, Ghayvat H (2021) Cnn variants for computer vision: history, architecture, application, challenges and future scope. Electronics 10(20):2470

Black S, Biderman S, Hallahan E, Anthony QG, Gao L, Golding L, He H, Leahy C, McDonell K, Phang J, Pieler MM. GPT-NeoX-20B: An Open-Source Autoregressive Language Model. InChallenges {\&} Perspectives in Creating Large Language Models

Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguist 5:135–146

Borgeaud S, Mensch A, Hoffmann J, Cai T, Rutherford E, Millican K, Van Den Driessche GB, Lespiau JB, Damoc B, Clark A, de Las Casas D (2022) Improving language models by retrieving from trillions of tokens. In: International conference on machine learning. PMLR, pp 2206–2240

Bosselut A, Rashkin H, Sap M, Malaviya C, Celikyilmaz A, Choi Y (2019) Comet: commonsense transformers for automatic knowledge graph construction. arXiv preprint arXiv:1906.05317

Bowman SR, Hyun J, Perez E, Chen E, Pettit C, Heiner S, Lukosiute K, Askell A, Jones A, Chen A, Goldie A (2022) Measuring progress on scalable oversight for large language models. arXiv preprint arXiv:2211.03540

Brown H, Lee K, Mireshghallah F, Shokri R, Tramèr F (2022) What does it mean for a language model to preserve privacy? In: Proceedings of the 2022 ACM conference on fairness, accountability, and transparency, pp 2280–2292

Bruch Marcel, Monperrus Martin, Mezini Mira (2009) Learning from examples to improve code completion systems. In: Proceedings of the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering, pp 213–222

Brundage M, Avin S, Clark J, Toner H, Eckersley P, Garfinkel B, Dafoe A, Scharre P, Zeitzoff T, Filar B, Anderson H (2018) The malicious use of artificial intelligence: forecasting, prevention, and mitigation. arXiv preprint arXiv:1802.07228

Burns C, Ye H, Klein D, Steinhardt J (2022) Discovering latent knowledge in language models without supervision. arXiv preprint arXiv:2212.03827

Cao Y, Li S, Liu Y, Yan Z, Dai Y, Yu PS, Sun L (2023) A comprehensive survey of ai-generated content (AIGC): a history of generative ai from gan to ChatGPT. arXiv preprint arXiv:2303.04226

Chen YC, Li L, Yu L, El Kholy A, Ahmed F, Gan Z, Cheng Y, Liu J (2020) Uniter: universal image-text representation learning. In: European conference on computer vision. Springer, pp 104–120

Chen X, Ding M, Wang X, Xin Y, Mo S, Wang Y, Han S, Luo P, Zeng G, Wang J (2022) Context autoencoder for self-supervised representation learning

Chen J, Guo H, Yi K, Li B, Elhoseiny M (2022) Visualgpt: data-efficient adaptation of pretrained language models for image captioning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 18030–18040

Chi Z, Huang S, Dong L, Ma S, Zheng B, Singhal S, Bajaj P, Song X, Mao XL, Huang H, Wei F (2022) Xlm-e: cross-lingual language model pre-training via electra

Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078

Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, Barham P, Chung HW, Sutton C, Gehrmann S, Schuh P (2022) Palm: scaling language modeling with pathways. arXiv preprint arXiv:2204.02311

Collobert R (2011) Deep learning for efficient discriminative parsing. In: Proceedings of the 14th international conference on artificial intelligence and statistics. JMLR workshop and conference proceedings, pp 224–232

Conneau A, Schwenk H, Barrault L, Lecun Y (2016) Very deep convolutional networks for text classification. arXiv preprint arXiv:1606.01781

Critch A, Krueger D (2020) AI research considerations for human existential safety (arches). arXiv preprint arXiv:2006.04948

Cui Y, Chen Z, Wei S, Wang S, Liu T, Hu G (2016) Attention-over-attention neural networks for reading comprehension. arXiv preprint arXiv:1607.04423

Curto G, Jojoa Acosta MF, Comim F, Garcia-Zapirain B (2024) Are AI systems biased against the poor? A machine learning analysis using Word2Vec and GloVe embeddings. AI Soc 39(2):617–632

Dadi Ramesh, Kumar Sanampudi Suresh (2022) An automated essay scoring systems: a systematic literature review. Artif Intell Rev 55(3):2495–2527

Dai Z, Yang Z, Yang Y, Carbonell J, Le QV, Salakhutdinov R (2019) Transformer-xl: attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860

Dai Z, Yang Z, Yang Y, Carbonell J, Le QV, Salakhutdinov RT (2019) Transformer-xl: attentive language models beyond a fixed-length context

Dalla-Torre H, Gonzalez L, Mendoza-Revilla J, Carranza NL, Grzywaczewski AH, Oteri F, Dallago C, Trop E, de Almeida BP, Sirelkhatim H, Richard G. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. BioRxiv. 2023 Jan 15:2023-01

Deng J, Lin Y (2022) The benefits and challenges of ChatGPT: an overview. Front Comput Intell Syst 2(2):81–83

Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805

Didi Y, Walha A, Wali A (2022) Covid-19 tweets classification based on a hybrid word embedding method. Big Data Cogn Comput 6(2):58

Dinan E, Humeau S, Chintagunta B, Weston J (2019) Build it break it fix it for dialogue safety: robustness from adversarial human attack. arXiv preprint arXiv:1908.06083

Ding N, Qin Y, Yang G, Wei F, Yang Z, Su Y, Hu S, Chen Y, Chan CM, Chen W, Yi J (2022) Delta tuning: a comprehensive study of parameter efficient methods for pre-trained language models. arXiv preprint arXiv:2203.06904

Dong X, Bao J, Zhang T, Chen D, Zhang W, Yuan L, Chen D, Wen F, Yu N, Guo B (2022) Peco: perceptual codebook for bert pre-training of vision transformers

Dong Y, Gu T, Tian Y, Sun C (2022) SNR: constraint-based type inference for incomplete java code snippets. In: Proceedings of the 44th international conference on software engineering, pp 1982–1993

Donghan Y, Zhu C, Yang Y, Zeng M (2022) Jaket: joint pre-training of knowledge graph and language understanding. Proc AAAI Confer Artif Intell 36:11630–11638

Dong L, Lapata M (2016) Language to logical form with neural attention. arXiv preprint arXiv:1601.01280

Dong L, Yang N, Wang W, Wei F, Liu X, Wang Y, Gao J, Zhou M, Hon HW (2019) Unified language model pre-training for natural language understanding and generation

Dufter P, Schmitt M, Schütze H (2022) Position information in transformers: an overview. Comput Linguist 48(3):733–763

Duque AB, Santos LL, Macêdo D, Zanchettin C (2019) Squeezed very deep convolutional neural networks for text classification. In: Artificial neural networks and machine learning—ICANN 2019: theoretical neural computation: 28th international conference on artificial neural networks, Munich, Germany, September 17–19, 2019, Proceedings, Part I, vol 28. Springer, pp 193–207

Everitt T, Lea G, Hutter M (2018) Agi safety literature review. In: Proceedings of the 27th international joint conference on artificial intelligence, pp 5441–5449

Fan A, Lewis M, Dauphin Y (2018) Hierarchical neural story generation. arXiv preprint arXiv:1805.04833

Fedus W, Zoph B, Shazeer N (2022) Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. J Mach Learn Res 23(1):5232–5270

Feng Y, Chen X, Lin BY, Wang P, Yan J, Ren X (2020) Scalable multi-hop relational reasoning for knowledge-aware question answering. arXiv preprint arXiv:2005.00646

Ferruz N, Schmidt S, Höcker B (2022) Protgpt2 is a deep unsupervised language model for protein design. Nat Commun 13(1):4348

Fu-Hao Yu, Chen K-Y, Ke-Han L (2022) Non-autoregressive ASR modeling using pre-trained language models for Chinese speech recognition. IEEE/ACM Trans Audio, Speech, Lang Process 30:1474–1482

Gage P (1994) A new algorithm for data compression. C Users J 12(2):23–38

Gan L, Teng Z, Zhang Y, Zhu L, Fei W, Yang Y (2022) Semglove: semantic co-occurrences for glove from bert. IEEE/ACM Trans Audio, Speech, Lang Process 30:2696–2704

Ganguli D, Askell A, Schiefer N, Liao TI, Lukošiūtė K, Chen A, Goldie A, Mirhoseini A, Olsson C, Hernandez D, Drain D (2023) The capacity for moral self-correction in large language models. arXiv preprint arXiv:2302.07459

Ganguli D, Hernandez D, Lovitt L, Askell A, Bai Y, Chen A, Conerly T, Dassarma N, Drain D, Elhage N, El Showk S (2022) Predictability and surprise in large generative models. In: 2022 ACM conference on fairness, accountability, and transparency, pp 1747–1764

Ghanem R, Erbay H (2023) Spam detection on social networks using deep contextualized word representation. Multimedia Tools Appl 82(3):3697–3712

Gidaris S, Komodakis N (2015) Object detection via a multi-region and semantic segmentation-aware CNN model. In: Proceedings of the IEEE international conference on computer vision, pp 1134–1142

Green C (1981) Application of theorem proving to problem solving. In: Readings in artificial intelligence, pp 202–222. Elsevier

Gu J, Cho K, Li VO (2017) Trainable greedy decoding for neural machine translation. arXiv preprint arXiv:1702.02429

Guo Xu, Yu Han (2022) On the domain adaptation and generalization of pretrained language models: a survey. arXiv preprint arXiv:2211.03154

Guo MH, Xu TX, Liu JJ, Liu ZN, Jiang PT, Mu TJ, Zhang SH, Martin RR, Cheng MM, Hu SM (2022) Attention mechanisms in computer vision: a survey. Comput Vis Media 8(3):331–368

Gurkaynak G, Yilmaz I, Haksever G (2016) Stifling artificial intelligence: human perils. Comput Law Secur Rev 32(5):749–758

Gururangan S, Lewis M, Holtzman A, Smith NA, Zettlemoyer L (2021) Demix layers: disentangling domains for modular language modeling. arXiv preprint arXiv:2108.05036

Han X, Zhang Z, Ding N, Yuxian G, Liu X, Huo Y, Qiu J, Yao Y, Zhang A, Zhang L et al (2021) Pre-trained models: past, present and future. AI Open 2:225–250

Han F, Zhang L, You X, Wang G, Li XY (2019) Shad: privacy-friendly shared activity detection and data sharing. In: 2019 IEEE 16th international conference on mobile ad hoc and sensor systems (MASS). IEEE, pp 109–117

He K, Chen X, Xie S, Li Y, Dollár P, Girshick R. Masked autoencoders are scalable vision learners. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2022 (pp. 16000-16009).

He, Pengcheng, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. “DEBERTA: DECODING-ENHANCED BERT WITH DISENTANGLED ATTENTION.” In International Conference on Learning Representations.

Hendrycks D, Basart S, Kadavath S, Mazeika M, Arora A, Guo E, Burns C, Puranik S, He H, Song D, Steinhardt J (2021) Measuring coding challenge competence with apps. arXiv preprint arXiv:2105.09938

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Hoffmann J, Borgeaud S, Mensch A, Buchatskaya E, Cai T, Rutherford E, Casas DD, Hendricks LA, Welbl J, Clark A, Hennigan T (2022) Training compute-optimal large language models. arXiv preprint arXiv:2203.15556

Holtzman A, Buys J, Du L, Forbes M, Choi Y (2019) The curious case of neural text degeneration. arXiv preprint arXiv:1904.09751

Hope T, Resheff YS, Lieder I (2017) Learning tensorflow: a guide to building deep learning systems. O’Reilly Media, Inc

Huo Y, Zhang M, Liu G, Lu H, Gao Y, Yang G, Wen J, Zhang H, Xu B, Zheng W, Xi Z (2021) Wenlan: bridging vision and language by large-scale multi-modal pre-training. arXiv preprint arXiv:2103.06561

Hwang JD, Bhagavatula C, Le Bras R, Da J, Sakaguchi K, Bosselut A, Choi Y (2021) (comet-) atomic 2020 on symbolic and neural commonsense knowledge graphs. Proc AAAI Confer Artif Intell 35:6384–6392

Jordan MI, Mitchell TM (2015) Machine learning: trends, perspectives, and prospects. Science 349(6245):255–260

Joulin A, Grave E, Bojanowski P, Mikolov T (2016) Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759

Kadavath S, Conerly T, Askell A, Henighan T, Drain D, Perez E, Schiefer N, Hatfield-Dodds Z, DasSarma N, Tran-Johnson E, Johnston S (2022) Language models (mostly) know what they know. arXiv preprint arXiv:2207.05221

Kalchbrenner N, Grefenstette E, Blunsom P (2014) A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188

Kaplan J, McCandlish S, Henighan T, Brown TB, Chess B, Child R, Gray S, Radford A, Wu J, Amodei D (2020) Scaling laws for neural language models. arXiv preprint arXiv:2001.08361

Keele S (2007) Guidelines for performing systematic literature reviews in software engineering (Vol. 5). Technical report, ver. 2.3 ebse technical report. ebse

Khan RA, Jawaid M, Khan AR, Sajjad M (2023) ChatGPT-reshaping medical education and clinical management. Pak J Med Sci 39(2):605

Kim Y, Denton C, Hoang L, Rush AM (2017) Structured attention networks. arXiv preprint arXiv:1702.00887

Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, Girshick R (2023) Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4015–4026

Korteling JH, van de Boer-Visschedijk GC, Blankendaal RA, Boonekamp RC, Eikelboom AR (2021) Human-versus artificial intelligence. Front Artif Intell 4:622364

Kowsher M, Sobuj MS, Shahriar MF, Prottasha NJ, Arefin MS, Dhar PK, Koshiba T (2022) An enhanced neural word embedding model for transfer learning. Appl Sci 12(6):2848

Kudo T (2018) Subword regularization: improving neural network translation models with multiple subword candidates. arXiv preprint arXiv:1804.10959

Kulikov I, Miller AH, Cho K, Weston J (2018) Importance of search and evaluation strategies in neural dialogue modeling. arXiv preprint arXiv:1811.00907

Kurdi G, Leo J, Parsia B, Sattler U, Al-Emari S (2020) A systematic review of automatic question generation for educational purposes. Int J Artif Intell Educ 30:121–204

Lample G, Sablayrolles A, Ranzato MA, Denoyer L, Jégou H (2019) Large memory layers with product keys. Adv Neural Inf Process Syst, 32

Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R (2019) Albert: a lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942

LeCun Y, Kavukcuoglu K, Farabet C (2010) Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE international symposium on circuits and systems. IEEE, pp 253–256

Lepikhin D, Lee H, Xu Y, Chen D, Firat O, Huang Y, Krikun M, Shazeer N, Chen Z (2020) Gshard: scaling giant models with conditional computation and automatic sharding. arXiv preprint arXiv:2006.16668

Letarte G, Paradis F, Giguère P, Laviolette F (2018) Importance of self-attention for sentiment analysis. In: Proceedings of the 2018 EMNLP workshop BlackboxNLP: analyzing and interpreting neural networks for NLP, pp 267–275

Levit M, Parthasarathy S, Chang S, Stolcke A, Dumoulin B (2014) Word-phrase-entity language models: Getting more mileage out of n-grams. In: 15th annual conference of the international speech communication association

Lewis M, Ghazvininejad M, Ghosh G, Aghajanyan A, Wang S, Zettlemoyer L (2020) Pre-training via paraphrasing. Adv Neural Inf Process Syst 33:18470–18481

Li LH, Yatskar M, Yin D, Hsieh CJ, Chang KW (2019) Visualbert: a simple and performant baseline for vision and language. arXiv preprint arXiv:1908.03557

Li J, Selvaraju R, Gotmare A, Joty S, Xiong C, Hoi SCH (2021) Align before fuse: vision and language representation learning with momentum distillation. Adv Neural Inf Process Syst 34:9694–9705

Liang W, Zou J (2022) Metashift: a dataset of datasets for evaluating contextual distribution shifts and training conflicts. arXiv preprint arXiv:2202.06523

Libbrecht P, Declerck T, Schlippe T, Mandl T, Schiffner D (2020) Nlp for student and teacher: concept for an ai based information literacy tutoring system. In: CIKM (workshops)

Li Z, Chen Z, Yang F, Li W, Zhu Y, Zhao C, Deng R, Wu L, Zhao R, Tang M, Wang J (2021) Mst: masked self-supervised transformer for visual representation

Lieber O, Sharir O, Lenz B, Shoham Y (2021) Jurassic-1: Technical details and evaluation. White Paper. AI21 Labs, 1(9)

Li J, Monroe W, Jurafsky D (2016) A simple, fast diverse decoding algorithm for neural generation. arXiv preprint arXiv:1611.08562

Lin BY, Chen X, Chen J, Ren X (2019) Kagnet: knowledge-aware graph networks for commonsense reasoning. arXiv preprint arXiv:1909.02151

Ling C, Zhao X, Lu J, Deng C, Zheng C, Wang J, Chowdhury T, Li Y, Cui H, Zhao T (2023) Beyond one-model-fits-all: a survey of domain specialization for large language models. arXiv preprint arXiv:2305.18703

Lin J, Yang A, Zhang Y, Liu J, Zhou J, Yang H (2017) A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130

Lin J, Yang A, Zhang Y, Liu J, Zhou J, Yang H (2020) Interbert: vision-and-language interaction for multimodal pretraining. arXiv preprint arXiv:2003.13198

Liu ZL, Dettmers T, Lin XV, Stoyanov V, Li X (2023) Towards a unified view of sparse feed-forward network in pretraining large language model. arXiv preprint arXiv:2305.13999

Liu J, Chang WC, Wu Y, Yang Y (2017) Deep learning for extreme multi-label text classification. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, pp 115–124

Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: a robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692

Liu L, Utiyama M, Finch A, Sumita E (2016) Neural machine translation with supervised attention. arXiv preprint arXiv:1609.04186

Li X, Yin X, Li C, Zhang P, Hu X, Zhang L, Wang L, Hu H, Dong L, Wei F, Choi Y (2020) Oscar: object-semantics aligned pre-training for vision-language tasks. In: Computer vision—ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16. Springer, pp 121–137

Lu J, Batra D, Parikh D, Lee S (2019) Vilbert: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. Adv Neural Inf Process Syst 32

Lu S, Guo D, Ren S, Huang J, Svyatkovskiy A, Blanco A, Clement C, Drain D, Jiang D, Tang D, Li G (2021) Codexglue: a machine learning benchmark dataset for code understanding and generation. arXiv preprint arXiv:2102.04664

Luong MT, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025

Luo R, Sun L, Xia Y, Qin T, Zhang S, Poon H, Liu TY. (2022) Biogpt: generative pre-trained transformer for biomedical text generation and mining. Br Bioinformat 23(6):bbac409

Lu J, Yang J, Batra D, Parikh D (2016) Hierarchical question-image co-attention for visual question answering. Adv Neural Inf Process Syst 29

Lv S, Guo D, Jingjing X, Tang D, Duan N, Gong M, Shou L, Jiang D, Cao G, Songlin H (2020) Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. Proc AAAI Confer Artif Intell 34:8449–8456

Maas A, Daly RE, Pham PT, Huang D, Ng AY, Potts C (2011) Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, pp 142–150

Madani A, Krause B, Greene ER, Subramanian S, Mohr BP, Holton JM, Naik N (2023) Large language models generate functional protein sequences across diverse families. Nat Biotechnol 41(8):1099–1106

Mahjour B, Hoffstadt J, Cernak T (2023) Designing chemical reaction arrays using phactor and ChatGPT. OPR&D 27(8):1510–1516

Manakul P, Liusie A, Gales MJ (2023) Selfcheckgpt: zero-resource black-box hallucination detection for generative large language models. arXiv preprint arXiv:2303.08896

Manna Z, Waldinger RJ (1971) Toward automatic program synthesis. Commun ACM 14(3):151–165

Manna Z, Waldinger R (1975) Knowledge and reasoning in program synthesis. Artif Intell 6(2):175–208

Mars M (2022) From word embeddings to pre-trained language models: a state-of-the-art walkthrough. Appl Sci 12(17):8805

Mbakwe AB, Lourentzou I, Celi LA, Mechanic OJ, Dagan A (2023) ChatGPT passing USMLE shines a spotlight on the flaws of medical education. PLOS digital health 2(2):e0000205

McCann B, Bradbury J, Xiong C, Socher R (2017) Learned in translation: Contextualized word vectors. Adv Neural Inf Process Syst 30

McLean S, Read GJ, Thompson J, Baber C, Stanton NA, Salmon PM (2023) The risks associated with artificial general intelligence: a systematic review. J Exp Theor Artif Intell 35(5):649–663

Mialon G, Dessì R, Lomeli M, Nalmpantis C, Pasunuru R, Raileanu R, Rozière B, Schick T, Dwivedi-Yu J, Celikyilmaz A, Grave E (2023) Augmented language models: a survey. arXiv preprint http://arxiv.org/abs/2302.07842

Mihaylov T, Frank A (2018) Knowledgeable reader: enhancing cloze-style reading comprehension with external commonsense knowledge. arXiv preprint http://arxiv.org/abs/1805.07858

Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. http://arxiv.org/abs/1301.3781

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst 26

Mikolov T, Sutskever I, Deoras A, Le HS, Kombrink S, Cernocky J (2012) Subword language modeling with neural networks. Preprint (http://www.fit.vutbr.cz/imikolov/rnnlm/char.pdf) 8(67):20

Min B, Ross H, Sulem E, Veyseh AP, Nguyen TH, Sainz O, Agirre E, Heintz I, Roth D (2021) Recent advances in natural language processing via large pre-trained language models: a survey. arXiv preprint arXiv:2111.01243

Mi H, Wang Z, Ittycheriah A (2016) Supervised attentions for neural machine translation. arXiv preprint arXiv:1608.00112

Nick B (2014) Superintelligence: paths, dangers, strategies, https://www.joyk.com/dig/detail/1608141862499156

Onitilo AA, Shour AR, Puthoff DS, Tanimu Y, Joseph A, Sheehan MT (2023) Evaluating the adoption of voice recognition technology for real-time dictation in a rural healthcare system: a retrospective analysis of dragon medical one. PLoS One 18(3):e0272545

Oubenali N, Messaoud S, Filiot A, Lamer A, Andrey P (2022) Visualization of medical concepts represented using word embeddings: a scoping review. BMC Med Inf Decis Mak 22(1):1–14

Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends® Inf Retr 2(1–2):1–135

Paranjape K, Schinkel M, Panday RN, Car J, Nanayakkara P (2019) Introducing artificial intelligence training in medical education. JMIR Med Educ 5(2):e16048

Parisotto E, Mohamed AR, Singh R, Li L, Zhou D, Kohli P (2016) Neuro-symbolic program synthesis. arXiv preprint arXiv:1611.01855

Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543

Petroni F, Rocktäschel T, Lewis P, Bakhtin A, Wu Y, Miller AH, Riedel S (2019) Language models as knowledge bases? arXiv preprint arXiv:1909.01066

Petukhova A, Matos-Carvalho JP, Fachada N (2024) Text clustering with llm embeddings. arXiv preprint arXiv:2403.15112

Pimpalkar A et al (2022) Mbilstmglove: embedding glove knowledge into the corpus using multi-layer bilstm deep learning model for social media sentiment analysis. Expert Syst Appl 203:117581

Press O, Smith NA, Lewis M (2021) Train short, test long: attention with linear biases enables input length extrapolation. arXiv preprint arXiv:2108.12409

Qiao C, Huang B, Niu G, Li D, Dong D, He W, Yu D, Wu H (2018) A new method of region embedding for text classification. In: ICLR (poster)

Qiu X, Sun T, Yige X, Shao Y, Dai N, Huang X (2020) Pre-trained models for natural language processing: a survey. Sci China Technol Sci 63(10):1872–1897

Radford A, Jeffrey W, Child R, Luan D, Amodei D, Sutskever I et al (2019) Language models are unsupervised multitask learners. OpenAI Blog 1(8):9

Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training, https://www.mikecaptain.com/resources/pdf/GPT-1.pdf

Rae JW, Borgeaud S, Cai T, Millican K, Hoffmann J, Song F, Aslanides J, Henderson S, Ring R, Young S, Rutherford E Hennigan T, Menick J, Cassirer A, Powell R, van den Driessche G, Hendricks LA, Rauh M, Huang P-S, Glaese A, Welbl J, Dathathri S, Huang S, Uesato J, Mellor J, Higgins I, Creswell A, McAleese N, Wu A, Elsen E, Jayakumar S, Buchatskaya E, Budden D, Sutherland E, Simonyan K, Paganini M, Sifre L, Martens L, Li XL, Kuncoro A, Nematzadeh A, Gribovskaya E, Donato D, Lazaridou A, Mensch A, Lespiau J-B, Tsimpoukelli M, Grigorev N, Fritz D, Sottiaux T, Pajarskas M, Pohlen T, Gong Z, Toyama D, de Masson d'Autume C, Li Y, Terzi T, Mikulik V, Babuschkin I, Clark A, de Las Casas D, Guy A, Jones C, Bradbury J, Johnson M, Hechtman B, Weidinger L, Gabriel I, Isaac W, Lockhart E, Osindero S, Rimell L, Dyer C, Vinyals O, Ayoub K, Stanway J, Bennett L, Hassabis D, Kavukcuoglu K, Irving G (2022) Scaling language models: methods, analysis & insights from training gopher

Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res 21(1):5485–5551

Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2023) Unsupervised broadcast news summarization; a comparative study on maximal marginal relevance (MMR) and latent semantic analysis (LSA). arXiv preprint arXiv:2301.02284

Rao A, Kim J, Kamineni M, Pang M, Lie W, Succi MD (2023) Evaluating ChatGPT as an adjunct for radiologic decision-making. medRxiv, pp 2023–02

Rao A, Pang M, Kim J, Kamineni M, Lie W, Prasad AK, Landman A, Dreyer KJ, Succi MD (2023) Assessing the utility of ChatGPT throughout the entire clinical workflow. medRxiv, pp 2023–02

Reed L, Li C, Ramirez A, Wu L, Walker M (2022) Jurassic is (almost) all you need: few-shot meaning-to-text generation for open-domain dialogue. In: Conversational AI for natural human-centric interaction: 12th international workshop on spoken dialogue system technology, IWSDS 2021, Singapore. Springer, pp 99–119

Reis ES, Costa CA, Silveira DE, Bavaresco RS, Righi RD, Barbosa JL, Antunes RS, Gomes MM, Federizzi G (2021) Transformers aftermath: current research and rising trends. Commun ACM 64(4):154–163

Ren H, Dai H, Dai B, Chen X, Yasunaga M, Sun H, Schuurmans D, Leskovec J, Zhou D. (2021) Lego: latent execution-guided reasoning for multi-hop question answering on knowledge graphs. In: International conference on machine learning. PMLR, pp 8959–8970

Ren S, He, K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst 28

Rives A, Meier J, Sercu T, Goyal S, Lin Z, Liu J, Guo D, Ott M, Zitnick CL, Ma J et al (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proc Natl Acad Sci 118(15):e2016239118

Robbes R, Lanza M (2008) How program history can improve code completion. In: 2008 23rd IEEE/ACM international conference on automated software engineering. IEEE, pp 317–326

Robinson J, Rytting CM, Wingate D (2022) Leveraging large language models for multiple choice question answering. arXiv preprint arXiv:2210.12353

Roller S, Sukhbaatar S, Weston J et al (2021) Hash layers for large sparse models. Adv Neural Inf Process Syst 34:17555–17566

Ross J, Belgodere B, Chenthamarakshan V, Padhi I, Mroueh Y, Das P (2022) Large-scale chemical language representations capture molecular structure and properties. Nat Mach Intell 4(12):1256–1264

Schwartz R, Dodge J, Smith NA, Etzioni O (2020) Green Ai. Commun ACM 63(12):54–63

Sahami M, Dumais S, Heckerman D, Horvitz E (1998) A Bayesian approach to filtering junk e-mail. In: Learning for text categorization: papers from the 1998 workshop, vol 62. Citeseer, pp 98–105

Sallam M (2023) The utility of ChatGPT as an example of large language models in healthcare education, research and practice: Systematic review on the future perspectives and potential limitations. MedRxiv 2023–02

Salmon PM, Carden T, Hancock PA (2021) Putting the humanity into inhuman systems: How human factors and ergonomics can be used to manage the risks associated with artificial general intelligence. Hum Factors Ergon Manuf Serv Industr 31(2):223–236

Salmon PM, Baber C, Burns C, Carden T, Cooke N, Cummings M, Hancock P, McLean S, Read GJ, Stanton NA (2023) Managing the risks of artificial general intelligence: a human factors and ergonomics perspective. Hum Factors Ergon Manuf Serv Industr 33(5):366–378

Samant RM, Bachute MR, Gite S, Kotecha K (2022) Framework for deep learning-based language models using multi-task learning in natural language understanding: a systematic literature review and future directions. IEEE Access 10:17078–17097

Saon G, Padmanabhan M (2001) Data-driven approach to designing compound words for continuous speech recognition. IEEE Trans Speech Audio Process 9(4):327–332

Sarker IH (2022) Ai-based modeling: techniques, applications and research issues towards automation, intelligent and smart systems. SN Comput Sci 3(2):158

Schick T, Schütze H (2021, April) Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp 255–269

Sennrich R, Haddow B, Birch A (2015) Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909

Shaghaghian S, Feng LY, Jafarpour B, Pogrebnyakov N (2020) Customizing contextualized language models for legal document reviews. In: 2020 IEEE international conference on big data (big data). IEEE, pp 2139–2148

Shaik T, Tao X, Dann C, Xie H, Li Y, Galligan L (2022) Sentiment analysis and opinion mining on educational data: a survey. Nat Lang Process J 2:100003

Shaw P, Uszkoreit J, Vaswani A (2018) Self-attention with relative position representations. arXiv preprint arXiv:1803.02155

Shen T, Mao Y, He P, Long G, Trischler A, Chen W (2020) Exploiting structured knowledge in text via graph-guided representation learning. arXiv preprint arXiv:2004.14224

Shen T, Zhou T, Long G, Jiang J, Pan S, Zhang C (2018) Disan: directional self-attention network for RNN/CNN-free language understanding. In: Proceedings of the AAAI conference on artificial intelligence, vol 32

Singh KN, Devi SD, Devi HM, Mahanta AK (2022) A novel approach for dimension reduction using word embedding: an enhanced text classification approach. Int J Inf Manag Data Insights 2(1):100061

Smith S, Patwary M, Norick B, LeGresley P, Rajbhandari S, Casper J, Liu Z, Prabhumoye S, Zerveas G, Korthikanti V, Zhang E (2022) Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. arXiv preprint arXiv:2201.11990

Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng AY, Potts C (2013) Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 conference on empirical methods in natural language processing, pp 1631–1642

Song Y, Wang J, Jiang T, Liu Z, Rao Y (2019) Attentional encoder network for targeted sentiment classification. arXiv preprint arXiv:1902.09314

Su J, Ahmed M, Lu Y, Pan S, Bo W, Liu Y (2021) Roformer: enhanced transformer with rotary position embedding. arXiv preprint arXiv:2104.09864

Subba B, Kumari S (2022) A heterogeneous stacking ensemble based sentiment analysis framework using multiple word embeddings. Comput Intell 38(2):530–559

Suhm B (1994) Towards better language models for spontaneous speech. In: Proc. ICSLP'94, vol 2, pp 831–834

Sukhbaatar S, Szlam A, Weston J, Fergus R (2015, December) End-to-end memory networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2, pp 2440–2448

Sun M, Xing J, Wang H, Chen B, Zhou J (2022) Mocl: data-driven molecular fingerprint via knowledge-aware contrastive learning from molecular graph

Sutskever Ilya, Martens James, Hinton Geoffrey E (2011) Generating text with recurrent neural networks. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp 1017–1024

Tai KS, Socher R, Manning CD (2015) Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075

Tai MC (2020) The impact of artificial intelligence on human society and bioethics. Tzu-Chi Med J 32(4):339

Tan H, Bansal M (2019) Lxmert: learning cross-modality encoder representations from transformers. arXiv preprint arXiv:1908.07490

Tang D, Qin B, Liu T (2016) Aspect level sentiment classification with deep memory network. arXiv preprint arXiv:1605.08900

Tan Q, Liu N, Huang X, Chen R, Choi SH, Hu X (2022) Mgae: Masked autoencoders for self-supervised learning on graphs. arXiv preprint arXiv:2201.02534

Tarasov DS (2015) Deep recurrent neural networks for multiple language aspect-based sentiment analysis of user reviews. In: Proceedings of the 21st international conference on computational linguistics dialog, vol 2, pp 53–64

Tay Y, Luu AT, Hui SC (2018) Hermitian co-attention networks for text matching in asymmetrical domains. IJCAI 18:4425–31

Tay Y, Dehghani M, Bahri D, Metzler D (2022) Efficient transformers: a survey. ACM Comput Surv 55(6):1–28

Thoppilan R, De Freitas D, Hall J, Shazeer N, Kulshreshtha A, Cheng HT, Le Q (2022) Lamda: Language models for dialog applications. arXiv preprint arXiv:2201.08239

Tian Y, Dong K, Zhang C, Zhang C, Chawla NV (2023) Heterogeneous graph masked autoencoders. Proceedings of the AAAI conference on artificial intelligence 37:9997–10005

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Kaplan Jared D, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell et al (2020) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901

Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, Roziére B, Goyal N, Hambro E, Azhar F, Rodriguez A (2023) Llama: open and efficient foundation language models. arXiv preprint arXiv:2302.13971

Tsai Y-S, Gasevic D (2017) Learning analytics in higher education—challenges and policies: a review of eight learning analytics policies. In: Proceedings of the 7th international learning analytics and knowledge conference, pp 233–242

Tsotsos JK, Culhane SM, Wai WY, Lai Y, Davis N, Nuflo F (1995) Modeling visual attention via selective tuning. Artif Intell 78(1–2):507–545

Vasileva O, Balyasnikova N (2019) Introducing vygotsky's thought: from historical overview to contemporary psychology. Front Psychol 10:1515

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30

Vijayakumar A, Cogswell M, Selvaraju R, Sun Q, Lee S, Crandall D, Batra D (2018, April) Diverse beam search for improved description of complex scenes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 32, No. 1

Waldinger RJ, Lee RC (1969) Prow: a step toward automatic program writing. In: Proceedings of the 1st international joint conference on artificial intelligence, pp 241–252

Wang Peifeng, Peng Nanyun, Ilievski Filip, Szekely Pedro, Ren Xiang (2020) Connecting the dots: a knowledgeable path generator for commonsense question answering. arXiv preprint arXiv:2005.00691

Wang SI, Manning CD (2012) Baselines and bigrams: simple, good sentiment and topic classification. In: Proceedings of the 50th annual meeting of the association for computational linguistics (volume 2: short papers), pp 90–94

Wang X, Kapanipathi P, Musa R, Mo Yu, Talamadupula K, Abdelaziz I, Chang M, Fokoue A, Makni B, Mattei N et al (2019) Improving natural language inference using external knowledge in the science questions domain. Proc AAAI Confer Artif Intell 33:7208–7215

Wang B, Liu K, Zhao J (2016) Inner attention based recurrent neural networks for answer selection. In: Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers), pp 1288–1297

Wang B, Shang L, Lioma C, Jiang X, Yang H, Liu Q (2021) On position embeddings in bert. In: International conference on learning representations

Wang G, Zhang L, Yang Z, Li XY (2018) Socialite: social activity mining and friend auto-labeling. In: 2018 IEEE 37th international performance computing and communications conference (IPCCC). IEEE, pp 1–8

Watanabe A, Wiseman SM (2023) A new era in surgical research: the evolving role of artificial intelligence. Am J Surg 226(6):923–925

Wei J, Tay Y, Bommasani R, Raffel C, Zoph B, Borgeaud S, Yogatama D, Bosma M, Zhou D, Metzler D, Chi EH (2022) Emergent abilities of large language models. arXiv preprint arXiv:2206.07682

Wei C, Wang YC, Wang B, Kuo CC (2023) An overview on language models: recent developments and outlook. arXiv preprint arXiv:2303.05759

Wen C, Hu Y, Li X, Yuan Z, Zhu XX (2023) Vision-language models in remote sensing: current progress and future trends. arXiv preprint arXiv:2305.05726

Wollny S, Schneider J, Di Mitri D, Weidlich J, Rittberger M, Drachsler H (2021) Are we there yet?–A systematic literature review on chatbots in education. Front Artif Intell 4:654924

Wu S, Irsoy O, Lu S, Dabravolski V, Dredze M, Gehrmann S, Kambadur P, Rosenberg D, Mann G (2023) Bloomberggpt: a large language model for finance. arXiv preprint arXiv:2303.17564

Xue H, Huang Y, Liu B, Peng H, Jianlong F, Li H, Luo J (2021) Probing inter-modality: visual parsing with self-attention for vision-and-language pre-training. Adv Neural Inf Process Syst 34:4514–4528

Xue L, Barua A, Constant N, Al-Rfou R, Narang S, Kale M, Roberts A, Raffel C (2022) Byt5: towards a token-free future with pre-trained byte-to-byte models. Trans Assoc Comput Linguist 10:291–306

Xue L, Constant N, Roberts A, Kale M, Al-Rfou R, Siddhant A, Barua A, Raffel C (2020) mt5: a massively multilingual pre-trained text-to-text transformer. arXiv preprint arXiv:2010.11934

Xu Z, Liu B, Wang B, Sun CJ, Wang X, Wang Z, Qi C (2017) Neural response generation via gan with an approximate embedding layer. In: Proceedings of the 2017 conference on empirical methods in natural language processing, pp 617–626

Yang YangAn, Wang Quan, Liu Jing, Liu Kai, Lyu Yajuan, Wu Hua, She Qiaoqiao, Li Sujian (2019) Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 2346–2357

Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV (2019) Xlnet: generalized autoregressive pretraining for language understanding. Adv Neural Inf Process Systems, 32

Yang J, Jin H, Tang R, Han X, Feng Q, Jiang H, Zhong S, Yin B, Hu X (2023) Harnessing the power of llms in practice: a survey on ChatGPT and beyond. arXiv preprint arXiv:2304.13712

Yang B, Tu Z, Wong DF, Meng F, Chao LS, Zhang T (2018) Modeling localness for self-attention networks. arXiv preprint arXiv:1810.10182

Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E (2016) Hierarchical attention networks for document classification. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 1480–1489

Yin Y, Chen C, Shang L, Jiang X, Chen X, Liu Q (2021) Autotinybert: automatic hyper-parameter optimization for efficient pre-trained language models. arXiv preprint arXiv:2107.13686

Zhang M, He Y (2020) Accelerating training of transformer-based language models with progressive layer dropping. Adv Neural Inf Process Syst 33:14011–14023

Zhang Y, Yang Q (2018) An overview of multi-task learning. Natl Sci Rev 5(1):30–43

Zhang Y, Ge C, Hong S, Tian R, Dong C, Liu J (2022) Delesmell: code smell detection based on deep learning and latent semantic analysis. Knowl-Based Syst 255:109737

Zhang C, D'Haro LF, Chen Y, Friedrichs T, Li H (2022) Investigating the impact of pre-trained language models on dialog evaluation. In: Conversational AI for natural human-centric interaction: 12th international workshop on spoken dialogue system technology, IWSDS 2021, Singapore. Springer, pp 291–306

Zhang Z, Han X, Liu Z, Jiang X, Sun M, Liu Q (2019) Ernie: enhanced language representation with informative entities. arXiv preprint arXiv:1905.07129

Zhang S, Roller S, Goyal N, Artetxe M, Chen M, Chen S, Dewan C, Diab M, Li X, Lin XV, Mihaylov T (2022) Opt: open pre-trained transformer language models. arXiv preprint arXiv:2205.01068

Zhao WX, Zhou K, Li J, Tang T, Wang X, Hou Y, Min Y, Zhang B, Zhang J, Dong Z, Du Y (2023) A survey of large language models. arXiv preprint arXiv:2303.18223

Zhao W, Zhu L, Wang M, Zhang X, Zhang J (2022) Wtl-CNN: a news text classification method of convolutional neural network based on weighted word embedding. Connect Sci 34(1):2291–2312

Zhao S, Zhang Z (2018) Attention-via-attention neural machine translation. In: Proceedings of the AAAI conference on artificial intelligence, vol 32

Zhao T, Zhao R, Eskenazi M (2017) Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. arXiv preprint arXiv:1703.10960

Zhou C, Li Q, Li C, Yu J, Liu Y, Wang G, Zhang K, Ji C, Yan Q, He L, Peng H (2023) A comprehensive survey on pretrained foundation models: a history from bert to ChatGPT. arXiv preprint arXiv:2302.09419

Zhu P, Qian T (2018) Enhanced aspect level sentiment classification with auxiliary memory. In: Proceedings of the 27th international conference on computational linguistics, pp 1077–1087

Zuccon G, Koopman B (2023) Dr ChatGPT, tell me what i want to hear: How prompt knowledge impacts health answer correctness. arXiv preprint arXiv:2302.13793

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.